Software Process Improvement (SPI)
Expert In
Small and Medium Scale Enterprises (SMEs)

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Engineering
in
Software Engineering

Submitted By
Nomi Baruah
(801031021)

Under Supervision of
Ms.Ashima Singh
Assistant Professor,CSED

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR UNIVERSITY
PATIALA – 147004

June 2012

CERTIFICATE

I hereby certify that the work which is being presented in the thesis report titled, *"Software Process Improvement(SPI) Expert in Small and Medium Scale Enterprises(SMEs)"*, in partial fulfillment of the requirements for the award of degree of Master of Engineering in *Software Engineering* submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of Ms.Ashima Singh and refers other researcher's works which are duly listed in the reference section.

The matter presented in this thesis has not been submitted for the award of any other degree of this or any other university.

Nomi Baruah

(Nomi Baruah)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

Ashima Singh

(Ms.Ashima Singh)
Assistant Professor
Computer Science & Engineering
Department,
Thapar University,Patiala.

Countersigned by

(Dr.Maninder Singh)
Associate Professor & H.O.D,
Computer Science & Engineering Department,
Thapar University,
Patiala.

( )
Dean (Academic Affairs),
Thapar University,
Patiala.

Acknowledgement

First of all, I am thankful to God for his blessings and for showing me the right direction. With his mercy, it has been made possible for me to reach so far.

It is a great privilege to express my gratitude and admiration towards my respected supervisor Ms.Ashima Singh . She has been an esteemed guide and a great support behind achieving the task. Without her able guidance, kind efforts and encouragement, the work wouldn't have been what it is. I am truly grateful to her for extending her total co-operation and understanding whenever I needed help and guidance from her.

I wish to express my heartiest thanks to Dr.Maninder Singh, Head, Department of Computer Science & Engineering, Thapar University, Patiala for providing me the opportunity and all necessary facilities to accomplish this thesis successfully.

I would also like to thanks my friends who were always there at the need of the hour and provided with the help for the completion of my thesis work. I am grateful to my parents who soulfully provided me their constant support, and encouraging attitude to undertake the challenge of this proportion.

I am also very thankful to the entire faculty and staff members of Computer Science & Engineering Department for their direct and indirect help, cooperation, love and affection which made my stay at Thapar University memorable.

Nomi Baruah

Nomi Baruah

(801031021)

Abstract

Software Process Improvement defines the identification of the current state-of-practice of processes within an organization and then improving it. Software Process Improvement is ever lasting, never ending and ever changing process. Software Process Improvement begins when a software development organization just begins to mature and it realizes the need to improve its processes. Some of the issues which force the organization to undergo software process improvement are customer dissatisfaction, inadequate software quality, inability to deliver on time and within budget, and excessive rework. A software organization regardless of their size wish to succeed on the market by providing software of high quality together with related services, support, communication with customers etc. Hence, small and medium enterprises (SMEs) can also dream of becoming a large organization in future if it can simply add standardized processes and work on software process improvement initiatives.

The small and medium software organizations do not have any specific software process improvement models. As their capacity to handle the cost of software improvement models like the CMM, P-CMM, SPICE and so forth are limited, they are not able to improve on their processes using SPI models. The SMEs continue to build product in random and unorganized manner which finally lead to a compromised product in quality.

SPI Expert in SMEs is an iterative model as well as an incremental model specially for small and medium software organizations.SPI Expert is a dedicated SPI Model for SMEs. SPI expert concentrates on the organization’s choice of processes those are given in support set. SPI expert contains SPI Expert Support Set which inherently is two staged model: The Basic Stage and the Elaborated Stage. These stages are actually materialized in given support set. And the SPI Expert Support Set is embedded in Process Flow. The two step approach will enable SMEs to improve and gain expertise in their process sets in stages.

Table of Contents

List of Figures

`List of Tables

Chapter 1
Introduction

1.1 Software Process Improvement

Though software had evolved through a long way, in most facets of modern life the historical problems are still unsolved. Process Improvement defines a way of looking how we can do our work in a better way. Software Process Improvement defines a way to improve the software processes in an organization in a practical way so that the organization meets business goals more effectively. The Software Process Improvement identifies the current state of practice of information systems within an organization and then improving on it [63]. To have an effective software process improvement, the organization needs to properly plan out and manage the work of product development.

A software organization, always wish to survive and succeed on the market by providing software of high quality .The small and medium scale enterprises (SME's) are striving for standardization of their software development processes and are carrying out improvement but not a proper set of processes. Software Process Improvement facilitates to the identification of current processes and to apply the changes to the current processes, so that the new processes can help in producing the high quality product [46]. To strengthen these small and medium scale enterprises, we need to improve the practices existing in the organization that have been adapted in accordance with their size and the type of business. It has been seen that only a few organization are able to achieve a high quality level for their development process. As most of the organizations in the world are small and medium scale enterprises, economy of a nation is greatly dependent on them.

1.2. Best Practices of Software Process Improvement

Some of the best practices of software process improvement are discussed below [67]. They are:

i. The organizations should have a working culture. The employees should know about the current working culture. Those organizations that have their own working culture have a positive approach for software process improvement.

ii. The organization should change their process, organizational structure and culture so that they will be able to cope up with the changing market demands and trends.

iii. The organizations should always align the software processes with business goals and objectives, because if the software processes are improved and it doesn't match with the business goals and objectives it is of no use.

iv. There should be a regular meeting between the software developers of the team to discuss about the ongoing project. There should be discussion about the current scenario of the project, the problems they had overcome since the last meeting and what problems they are facing with.

v. The Software Process Improvement should always be treated like a project.

vi. Always test the new software process using a trailblazer/pilit project to validate it.

vii. Introduce a Software Engineering Process Group (SEPG) in your organization so that they have the sole responsibility to support the definition and improvement in your organization software process.

viii. Recruit staff for SEPG with actual practitioners because the people who best understand how to develop the software are the people who are very good at developing the software.

ix. Bring an expert to the organization that has both a process background and an Object-oriented development background so that we can increase the chance of success of software process improvement in the organization.

x. A user guide should be developed for the process so that the staff can learn about the chosen processes as it is going to be implemented in the organization.

1.3 Need of a Software Process Improvement Model

Many of the projects fail in an organization because they were not on schedule, were subject to cost overruns or did not meet their desired quality standards. One of the major reasons these software projects can spin out of control is a lack of proper planning. The solution to many of these problems stems from having a well-defined, project-specific process in place to ensure adequate planning and communication occur at the start of each project, which increases the probability of success for a project.

In order to help practitioners to stay on target with software development tools, different software process models are being proposed.

Software Process Models are being proposed by taking the abstracts of software processes. Software Process are structured set of activities required to develop a software system i.e. specification, design, validation and evolution [5]. The definition of a software process usually specifies the actors executing the activities, their roles and the artifacts produced. An organization can define on its own way to produce software.

Curtis et al.present some of the specific goals and benefits of modeling the software process [8]:

i. Ease of understanding and communication: requiring a process model containing enough information for its representation. It formalizes the process, thus providing a basis for training.
ii. Process management support and control: requiring a project-specific software process and monitoring, management and co-ordination.
iii. Provision for automated orientations for process performance: requiring an effective software development environment, providing user orientations, instructions and reference material.
iv. Provision for automated execution support: requiring automated process parts, co-operative work support, a compilation of metrics and process integrity assurance.
v. Process improvement support: requiring the reuse of well defined and effective software processes, the comparison of alternative processes and process development support.

Different process models can define their perspective points of view at a different tangent. The software process basically consists of two interrelated processes: the production process and the management process.
The first is related to the construction and maintenance of the software product, whereas the second is responsible for estimating, planning and controlling the necessary resources (people, time, technological, ...) to be able to carry out and control the production process [24].

Different elements of a process, e.g. activities, products (artifacts), resources (personnel and tools) and roles, can be modeled. The elements most commonly modeled are presented below. They are Agent or Actor, Role, Activity and Artifact or Product[30].The fig 1.1 shows the relationship between the elements.

1.4 Scope of Study

It has been seen that the different SMEs strive to improve their productivity and quality of their product. They are using SPI models but due to lack of cost, resources they are not able to follow

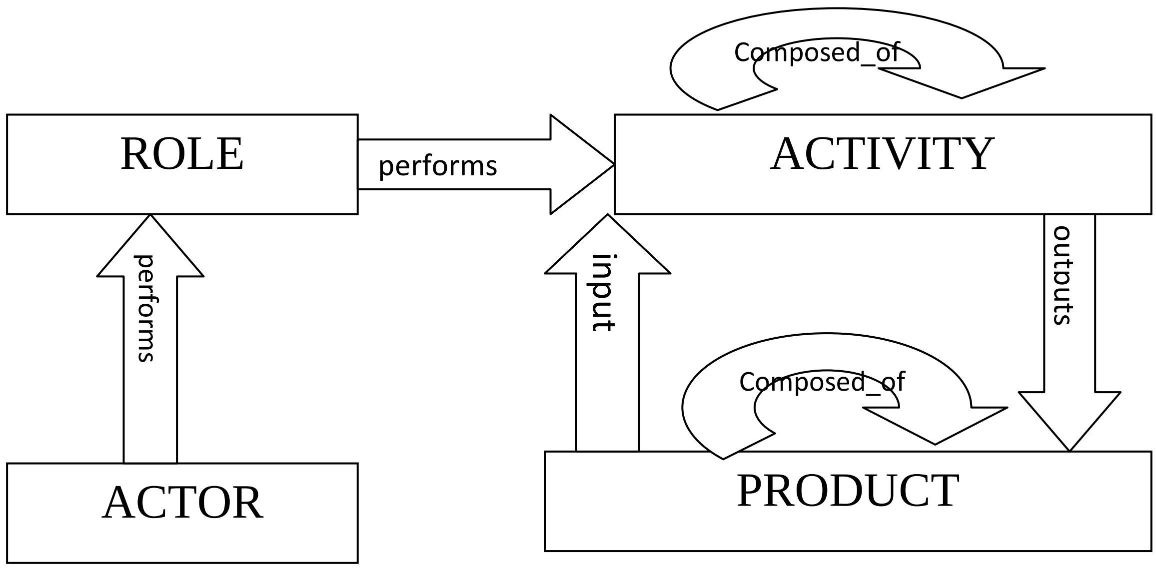

Fig 1.1: Basic Process Modeling Components [30]

all the processes i.e. the SMEs don't follow any structured models for product as well as process improvisation. This study finds its scope in small and medium organizations those are really striving for a cost effective process improvement model. SMEs in Indian software industry needs SPI model that can support already followed processes in their organization while integrating new customized processes.

1.5 Gap Analysis

Software process improvement is ever changing, ever evolving and never ending process. Especially SMEs have not yet found any SPI model that can cater according to their pocket. Such organizations are performing excellent in market but unable to retain due to lack of effective process set. Therefore, customer dissatisfaction, inadequate software quality, inability to deliver on time and within budget and excessive rework are its direct or indirect results.

1.6 Objectives of the Thesis

The objectives of the thesis are described below. They are

i. To study Software Processes and Software Process Improvement Models,

ii. To study Agile Software Development Methodologies.

iii. To conduct a survey in different Indian SMEs in order to find out the different trends and strategies followed by them.

iv. To Analyze and tabulate the results of the conducted survey.

v. To propose a Model.

1.7 Organization of the Thesis

Chapter 2 summarizes the literature survey carried out on various software process models and software process improvement models. Agile software development methodologies are discussed also.

Chapter 3 studies the problem identified in Software Process Improvement faced by small and medium organizations.

Chapter 4 analyses the facts and findings discovered in the survey conducted in eighteen small and medium software organizations of India.

Chapter 5 illustrates the SPI Expert in SMEs, the proposed model enriched with effective processes as a part of SPI Expert Support Set.

Chapter 6 summarizes the conclusions and future scope.

Chapter 2

Software Process Improvement

2.1 Software Process Improvement Models

2.1.1. Capability Maturity Model (CMM)

The Software Engineering Institute (SEI) had created a Capability Maturity Model (CMM) for developing and maintaining software so that a proper management of software processes can be done in an organization. The model helps the organization to evolve towards a culture of software engineering and management culture. It also guides the organization how to proceed for software process improvement (SPI) strategies [35].

The SEI has developed a five-level Capability Maturity Model for software that provides organizations with guidance for measuring software process maturity and establishing process improvement plans. These five maturity levels define an ordinal scale for measuring the maturity of an organization's software process and for evaluating its software process capability [36].

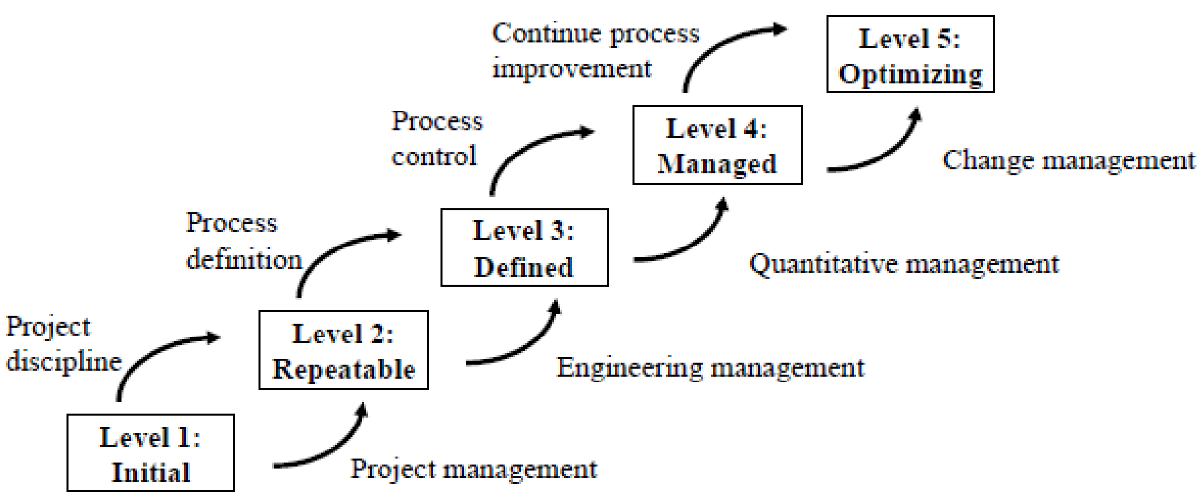

Fig 2.1.1: Maturity Levels of CMM [61]

2.1.2. Personal Software Process (PSP)

The personal software process (PSP) is developed to provide the software engineers a way to improve the quality, predictability, and productivity of their work. It is designed to address the improvement needs of individual engineers and small software organizations.

The PSP process phase has three phases. They are planning, development and post mortem [33]. The approach was developed such that it addresses the need in software process improvement of small and medium scale enterprises. The PSP uses various standard software engineering methods, so that it can show the engineers how a defined and measured process can help them to improve their personal performance.

Maturity Levels	Key Process Areas
Level 5-Optimizing	Process Change Management Technology Change Management Defect Prevention
Level 4-Managed	Quality Management Quantitative Process Management
Level 3-Defined	Peer Reviews Software Product Engineering Integrated Software Management Software Process Definition Software Process Focus
Level 2-Repeatable	Software Project Tracking and Oversight Software Project Planning
Level 1-Initial	

Fig 2.1.2: The PSP [62]

2.1.3 Team Software Process (TSP)

The TSP is designed such that the team as a whole is convinced to increase the effectiveness of the method they will use to increase the quality the product they will make. The TSP process is defined in a series of process scripts in a way so that it will describe all aspects of project planning and product development. TSP concentrates on the people aspect of the software triangle. It emphasizes the fact that “Together Everyone Achieves More” [40]. The principal benefit of TSP is that it shows engineers that working in a team, how to produce quality products for planned costs and on aggressive schedule by managing their work and owning their own processes and plans [65].

2.1.4 Capability Maturity Model Integration (CMMI)

CMMI is a process improvement approach whose aim is to improve the usability of maturity models by integrating many different models in one framework. CMMI improves the performance of an organization with essential elements of effective processes, addresses productivity, performance, costs and stakeholder satisfaction. CMMI is applied to teams, work groups, projects, divisions and entire organizations [32].

The representation of CMMI can be staged or continuous [32]. The user can focus on the specific processes through continuous representation of CMMI, which are considered important for the organization's immediate business objectives, or those to which the organization assigns a high degree of risk.

2.1.5 Six-Sigma (SS)

Six-Sigma is represented as “6σ”.Six Sigma has emerged as one of the most effective business improvement strategies over the years. It is the methodology having statistical base focusing on removing causes of variations or defects in the product or core business processes. The main aim of this approach is to improve efficiency and effectiveness of the organization. Six Sigma is a statistical term that refers to 3.4 Defects per Million Opportunities (DPMO) (or 99.99966 percent accuracy) [10]. A defect can be anything from a faulty part to an incorrect customer bill. When customer dissatisfaction occurs, defect results. SMEs have a much closer proximity to customers. Six Sigma is the current stage of evolution in the field of Quality management with a core focus on profitability improvement and strategic value levelling. The methodology is targeted on the processes within an organization.

Six Sigma is based on DMAIC process. D stands for define, M stands for measure, A stands for analysis, I stands for improve and C stands for control. In define phase, the problem is being stated [19]. In measure phase, the process performance is estimated, so that the present sigma level can be find out. In analysis phase, opportunities for improvement are explored. In improve phase, recommended solutions that are being explored in analysis phase, are being implemented at this phase. Lastly in Control phase, the deviations from the recommended method to the actual progress and reasons for deviations will be analyzed. The organizational infrastructure of Six-

Sigma consists of different levels of experts. They are Champions, Master Black Belts, Black Belts, Green Belts and Team Members [18].

2.1.6. People-CMM (P-CMM)

People Capability Maturity Model (P-CMM): The P-CMM is developed to provide a framework that continuously develops the human assets of the software or information systems organization. The P-CMM is an organizational change model. It is through P-CMM that an organization is able to identify the capabilities of current workforce practices and also able to evolve the workforce practices with the help of maturity levels of P-CMM. The P-CMM is constructed from workforce practices and process improvement techniques that have proven effective in many organizations. [9] The range of organizational processes that the P-CMM addresses is extensive and covers areas of workforce management. The P-CMM model is divided into five maturity levels .Each level in this model defines a new standard for people management practices. The first level is the initial state. The concept of second level is that a practice cannot be improved if it cannot be repeated. Institutionalizing the processes that work best is the core for the third level. Once the organization can adjust its process with process data it will end at level four. The fifth level is a stage where the organization focuses on continuous process improvement

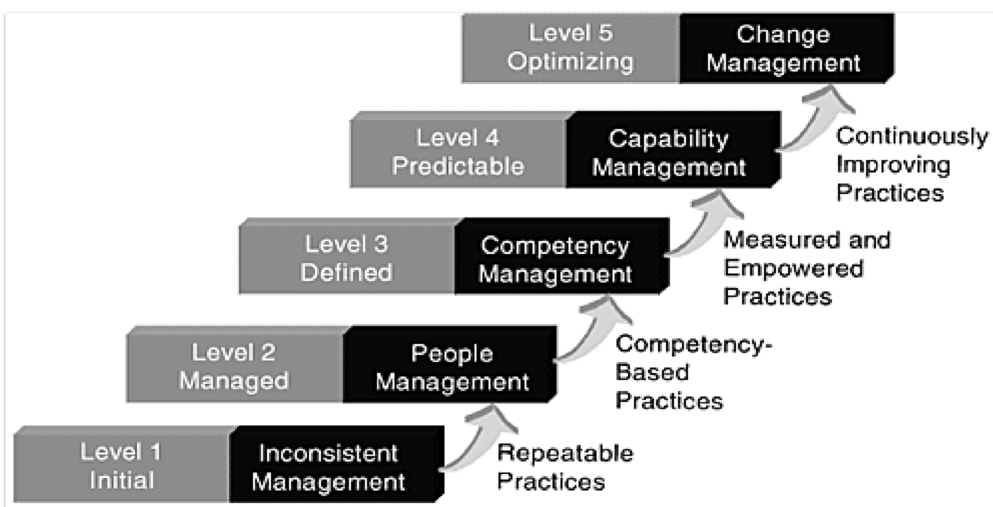

Fig 2.1.3: Maturity level of P-CMM [44]

2.1.7. Malcolm Balridge National Quality Award (MBNQA)

The Malcolm Baldrige National Quality Improvement Act of 1987 establishes the Malcolm Balridge National Quality Award. The award recognizes quality and performance excellence of organizations. The award is awarded by the U.S President to U.S organizations. The U.S. Commerce Department's National Institute of Standards and Technology manages the award and ASQ administers it [66].

The Malcolm Balridge National Quality Award is given to organizations based on achievement and improvement in seven areas known as Balridge Criteria for Performance Excellence. The seven areas are leadership, strategic planning, customer and market focus, measurement, analysis, and knowledge management, human resource focus, process management and business/organizational performance results [64].

2.1.8. System Engineering CMM (SE-CMM)

SE-CMM: SE-CMM stands for System Engineering Capability Maturity Model. The Systems Engineering CMM effort began in August 1993 in response to industry requests for assistance in coordinating and publishing a model for the systems engineering community analogous to the CMM for Software [49].The main goal of SE-CMM is to improve system or product engineering process. The SE-CMM is becoming a reference model for assessing current practices, planning, implementing, and measuring process improvement efforts and determining the business results of such efforts [49].

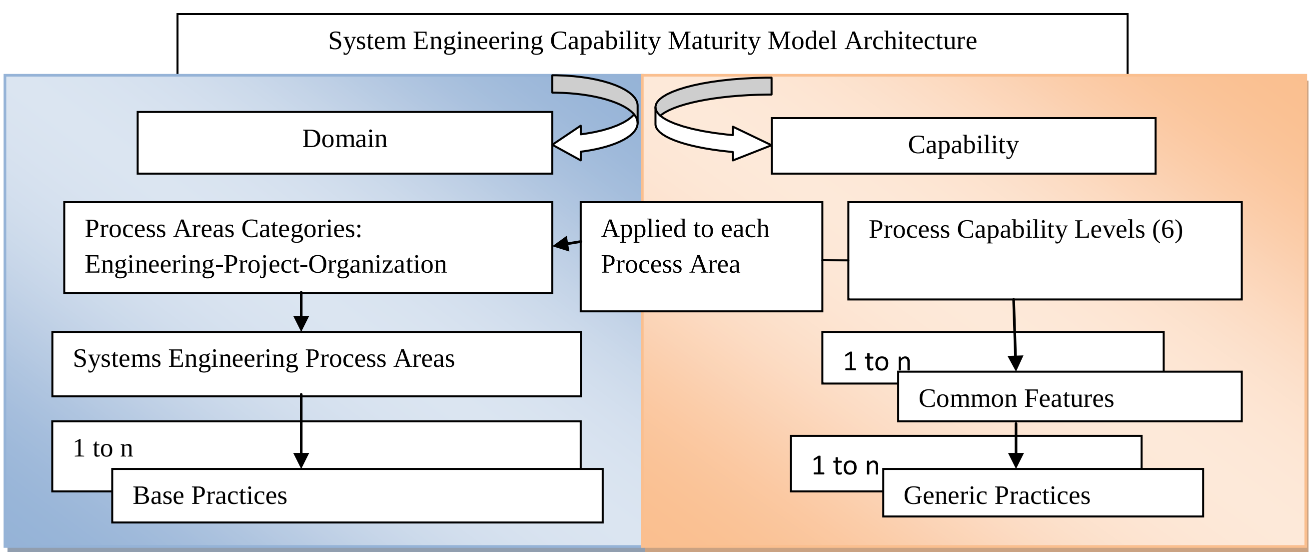

Fig 2.1.4: SE-CMM Model Architecture [49]

2.1.9. System Security Engineering CMM (SSE-CMM)

The SSE-CMM is developed from SE-CMM. The main objective of SSE-CMM model is to provide a defined, mature and measurable discipline in advance security engineering [55]. SSE-CMM is a distinct model that improves and assesses security engineering capability with integration with other engineering capability. The SSE-CMM integrates with all other engineering disciplines such as systems, software, hardware, and human factors. All organizations can use SSE-CMM model [15].

2.1.10. ISO 9000

The International Standards Organization proposed a general quality standard ISO 9000.The standard is applicable to manufacturing industry, service and software industries. The ISO 9000 consist of a family of standards which is being assisted in different organizations, consisting of all types and sizes so that effective quality management systems can be implemented and operated [22]. ISO 9000 is often called as a quality system or a quality management system. The ISO 9000 follow three broad quality concepts. They are in order to meet the continually stated or implied needs of the customer an organization should achieve and sustain the quality of product or service produced, quality practices should be tracked in an organization so that to achieve and sustain intended quality, a confidence is provided to the management and the purchaser should receive a confidence from the organization that the intended quality is being, or will be, achieved in the delivered product or service provided [22].

2.1.11. Trillium Model

The Trillium Model has been developed from a customer perspective, as perceived in a competitive, commercial environment. The Trillium model has been initially designed for embedded software systems but the model is being applied to other segments of software industry such as Management Information Systems (MIS). Trillium has been developed by a consortium of telecommunications companies headed by Bell Canada. Trillium model is especially suited for those organizations who want to conduct software improvement programs on existing processes [16]. The Trillium model uses the road-map concept from which sets of software process improvement goals can be defined. . The Trillium model is quite similar to the CMM [52]. The key process areas of the CMM have been condensed into fewer capability areas that are elaborated and expanded into various roadmaps.

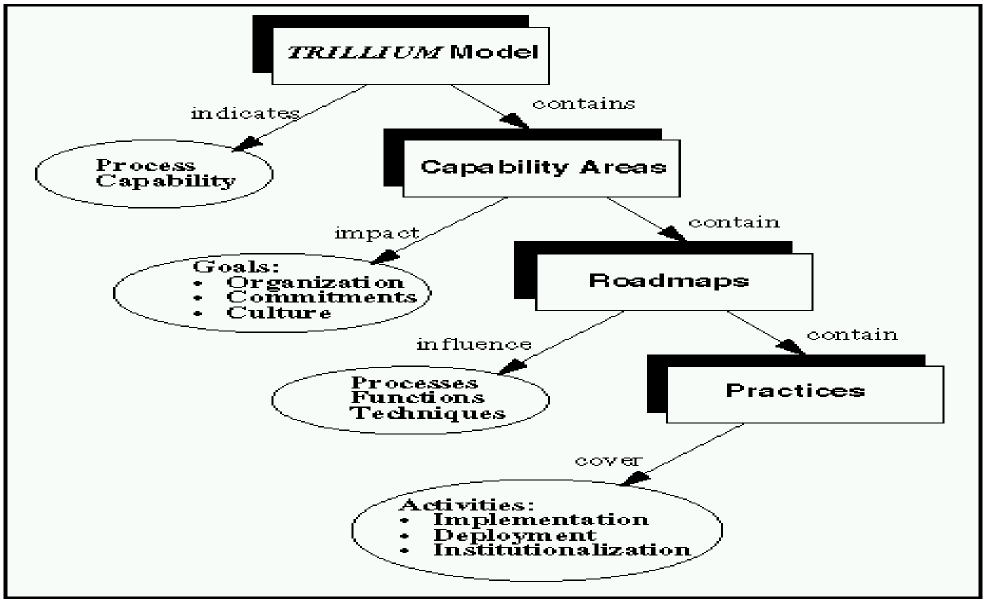

Fig 2.1.5: Architecture of the Trillium Model [58]

2.1.12. SPICE (Software Process Improvement Capability dEtermination)

The SPICE was developed as a joint result of International Organization for Standardization and the International Electrotechnical Commission. The model was an international standard on process assessment. SPICE is also known as ISO/IEC 15504.The main objective behind its development is to provide the software industry with gains in productivity and quality. The model has three potential users [4].

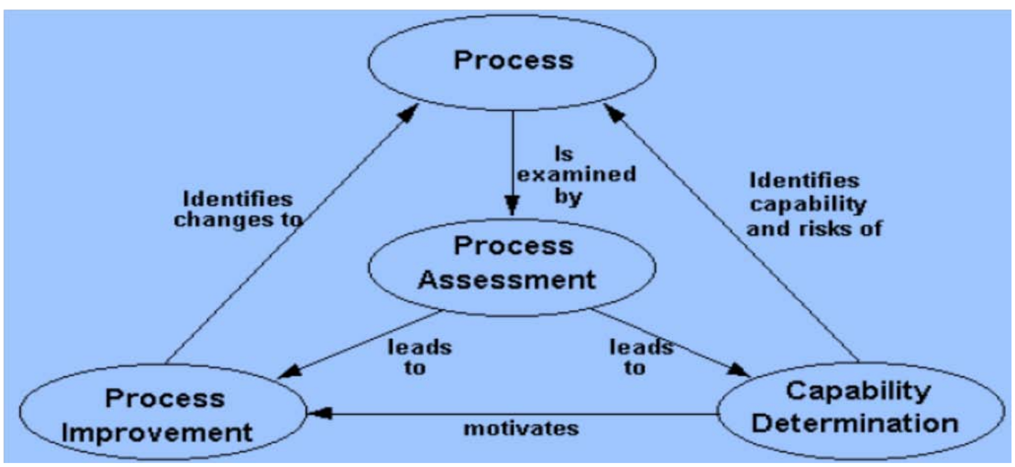

Fig 2.1.6: SPICE [50]

They are purchasers, acquirers and software suppliers. The SPICE provides a framework and set of requirements that can accommodate a variety of models of software practice and assessment methods [50].Capability dEtermination reduces the risk associated with large projects and at the same time helps purchasers to get better value for money. [14] SPICE follows a continuous architecture for capability. Thus each process category is rated separately on a scale from 0 to 5.The Fig 2.1.6 describes the architecture of SPICE

2.1.13. ISO/IEC 12207

ISO/IEC 12207 was developed in 1989 for software life cycle processes to fill a critical need. The critical need is that ISO/IEC had provided a common framework for the software practitioners so that they get an environment for developing and managing software. There exist 17 process areas (PAs) in ISO/IEC 12207.The process areas are divided into modularity and responsibility [47]. The life cycle of the processes are divided into three categories. They are primary, supporting and organizational. The Fig 2.1.7 describes the different process areas of ISO/IEC 12207.

1.Primary Process Areas	2.Supporting Process Areas	3.Organizational Process areas
1.1Acquisition 1.2.Supply 1.3.Development 1.4.Operation 1.5Maintenance	2.1.Documenation 2.2.Configuration Management 2.3.Quality Assurance 2.4.Verification 2.5.Validation 2.6.Joint Review 2.7.Audit 2.8.Problem Resolution	3.1.Management 3.2.Infrastructure 3.3.Improvement 3.4.Training

Fig 2.1.7: ISO/IEC 12207 [47]

2.1.14. BOOTSTRAP

BOOTSTRAP is a software process improvement and assessment methodology. The development of BOOTSTRAP was especially for European industries. The main goal of BOOTSTRAP development is to speed up the use of technology in European software industry [43]. The Fig 2.1.8 of BOOTSTRAP is designed in a tree structure. It contains process categories, processes and best practices. The unique characteristic of this model is that, software processes and product of different types can use this methodology.

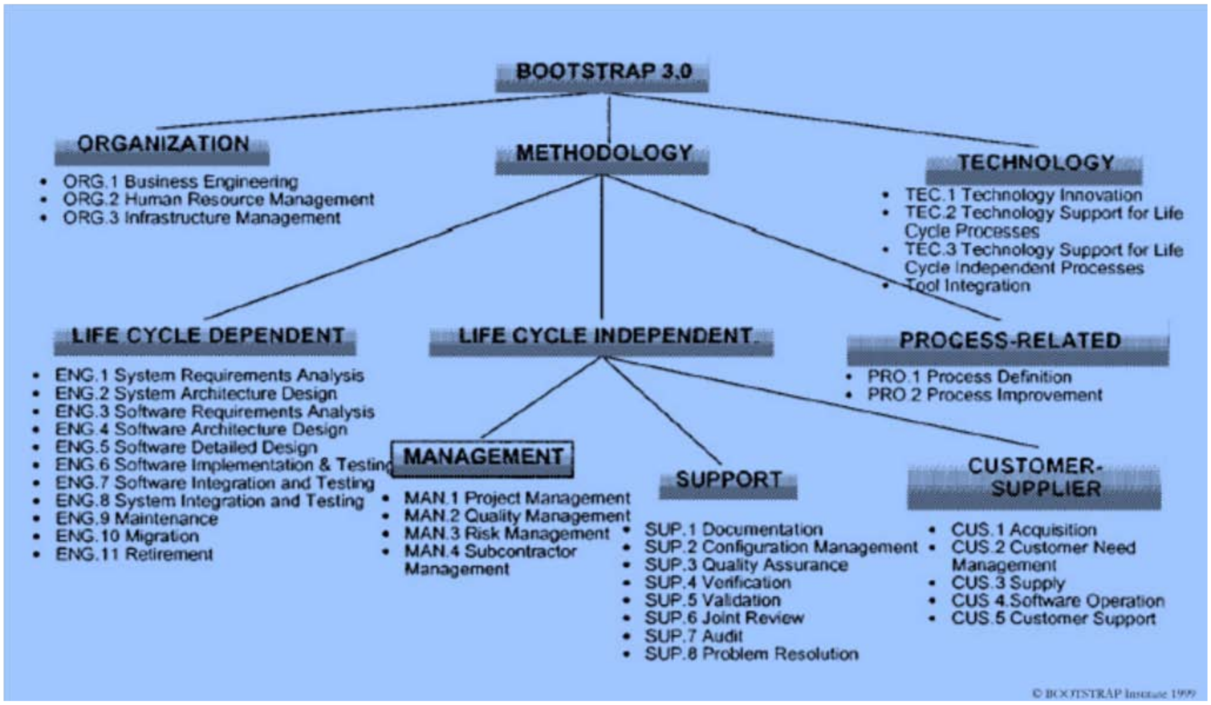

Fig 2.1.8: BOOTSTRAP [43]

The methodology helps the software developers to identify the strength and weakness of the processes and support in software improvement planning so that organizational goals are being achieved and the quality of the products increases [43]. A new release (Release 3.0) of the BOOTSTRAP methodology has been developed to assure conformance with the emerging ISO standard for software process assessment and improvement. The main objective of this methodology was to aid adoption of software engineering technology in the small and medium sized organizations in the European software industry [59].

2.1.15. SECAM (Systems Engineering Capability Assessment Method)

SECAM is an assessment model for organization that includes processes and non-processes indicators for capability. The model describes performance of an organization with two elements. They are capability and capacity. All system engineering capability are being assessed by SECAM.SECAM was also referred to as “interim model” during its early phases of development which is being dropped during later stages of development. The organizations can use SECAM model, which develop products or provide services representative. SECAM consist of 19 key focus areas. The key focus areas of SECAM are divided into three categories: Management, Organization and Systems Engineering [21].

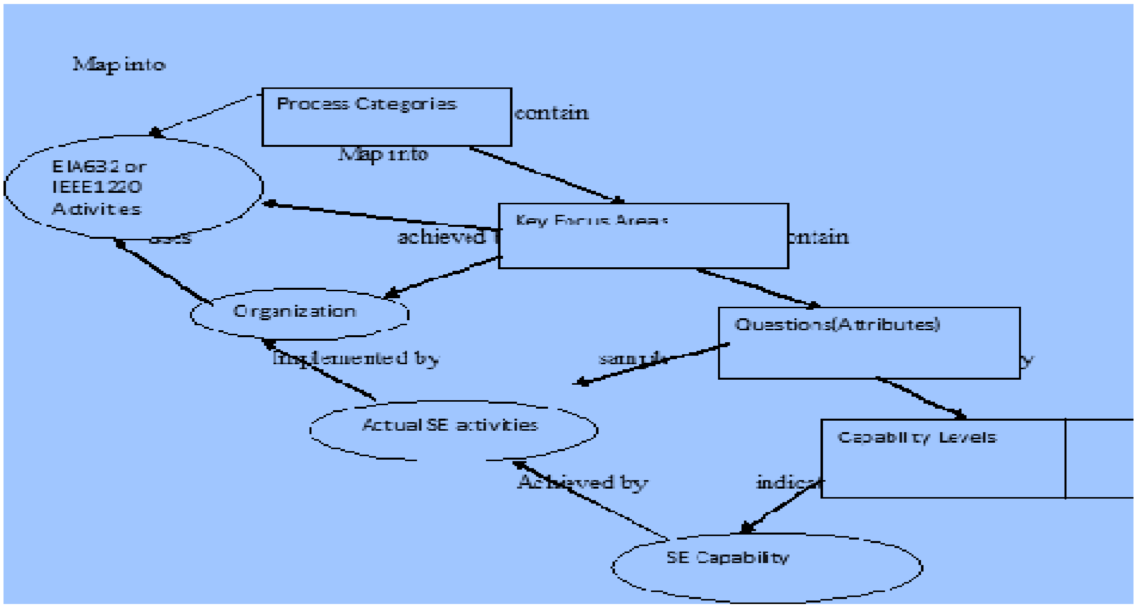

Fig 2.1.9: Structure of the INCOSE SECAM [21]

2.1.16. SDCE (Software Development Capability Evaluation)

The Software Development Capability Evaluation is a structured methodology developed by an Air Force-sponsored Process Action Team [14]. The SDCE is a standardized approach composed of questions and related criteria for assessing the maturity of each software development contractor for a specific program. The structure of SDCE model comprises of functional areas, critical capability areas, critical capabilities, and questions and criteria associated with program management and systems and software engineering processes. The process areas of SDCE are sub-divided into six sub-categories. They are Program Management, System Engineering, Software Engineering, Quality Management and Product Control, Organizational Resources and Program Support and Program Specific Technologies [14].

2.1.17. MIL-STD-498

MIL-STD-498 was established so that a uniform requirements are being established for software development and documentation. MIL-STD-498A was released after MIL-STD-498.Then MIL-STD-498B was developed which was never released. From MIL-STD-498B, two new models were developed and released. They are ANSI/EIA 632 and IEEE 1220 [48]. The standard is being used for performing software development and can be applied to contractors, subcontractors, or government in-house agencies. For establishing uniform requirements, three terms are being used. They are acquirer, developer and contract. Organization which requires technical effort, the term acquirer is being addressed, organization which performs the technical effort, the term developer is being addressed and, the term contract is being addressed for the agreement between these parties [67].

2.1.18. IEEE 1220

IEEE 1220 is a system engineering life cycle standard. IEEE 1220 conforms to the requirements of ISO 12207.IEEE 1220 describes generic, problem-solving systems engineering processes that produce the specifications, baselines and related products for project development. The model resembles the incremental (iterative) waterfall model [56]. The IEEE 1220 process model is described in the figure on next page.

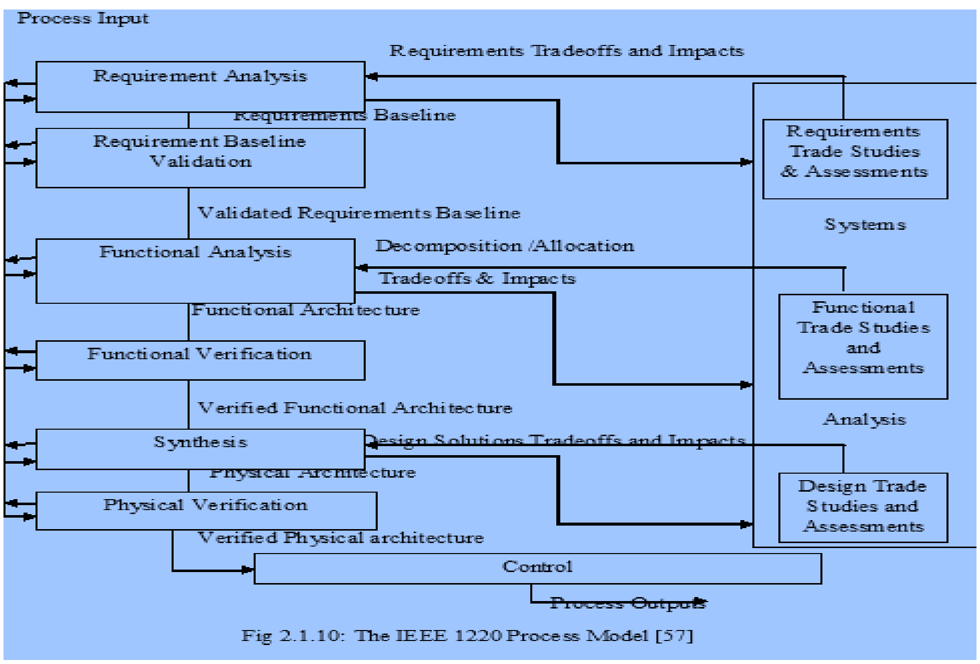

Fig 2.1.10: The IEEE 1220 Process Model [57]

2.1.19. TickIT

The TickIT certification scheme was developed by the Department of Trade and Industry(DTI) and the British Computer Society (BCS) with strong support of UK software suppliers .An organization’s quality management system(QMS) are being addressed by TickIT certification scheme which are concerned with the specification, design, development, installation and support of computer software [57].

TickIT certification is an ongoing process. A 6-monthly surveillance visit is done in TickIT certification. After every third year recertification is done in an organization for TickIT certification.TickIT certification acts as an indicator for organizations, which are between level 2 and level 3 of the maturity scale, as defined in CMM [1]. The figure below shows that the TickIT certification is being doubled every year.

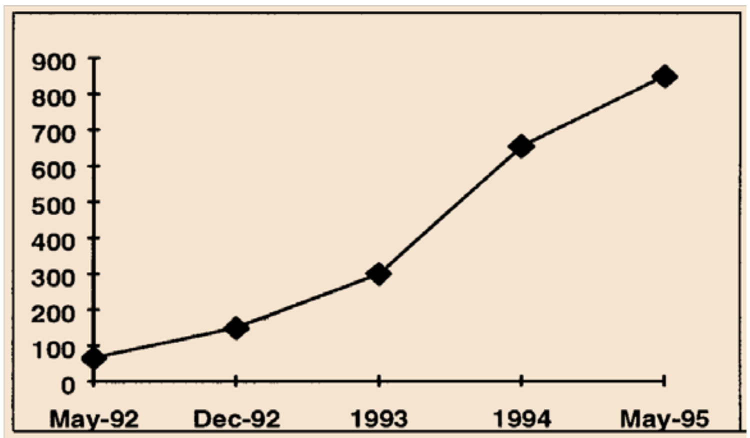

Fig 2.1.11: Number of Organization with TickIT certification [1]

2.1.20. IDEAL Model

The IDEAL model is not constant for all the organization. The model depends on the resources, a particular organization can commit for the software process improvement(SPI) program [11].

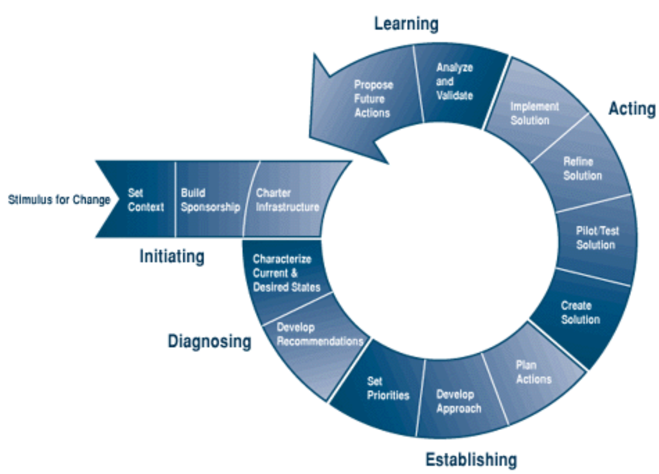

Fig 2.1.12: The IDEAL Model [27]

The Fig 2.1.12 describes the different phases of IDEAL Model. The initializing phase defines the infrastructure of software process improvement, roles and responsibilities and resources. The SPI plan is created in this phase. The path of continuous process improvement starts from diagnosing phase. The establishing phase puts into action the different areas for software process improvement. In acting phase, the different areas that are implemented for software process improvement, risks and barriers are identified that became an obstacle to software process improvement. In learning phase, it is tried to not repeat the same .

2.1.21. PRISMS

Process Improvement for Small to Medium enterprises (PRISMS).PRISMS is an action research project, with a team of three researchers from Leeds Metropolitan University. The model enables individual SMEs to tailor the software process improvement to the organization's business objectives. PRISMS assessment consists of an awareness and business case workshop which focuses on process improvement, cost-effectiveness, and providing a road-map for process improvement, followed by a series of assessment interviews where we meet different groups individually and collectively. [45]

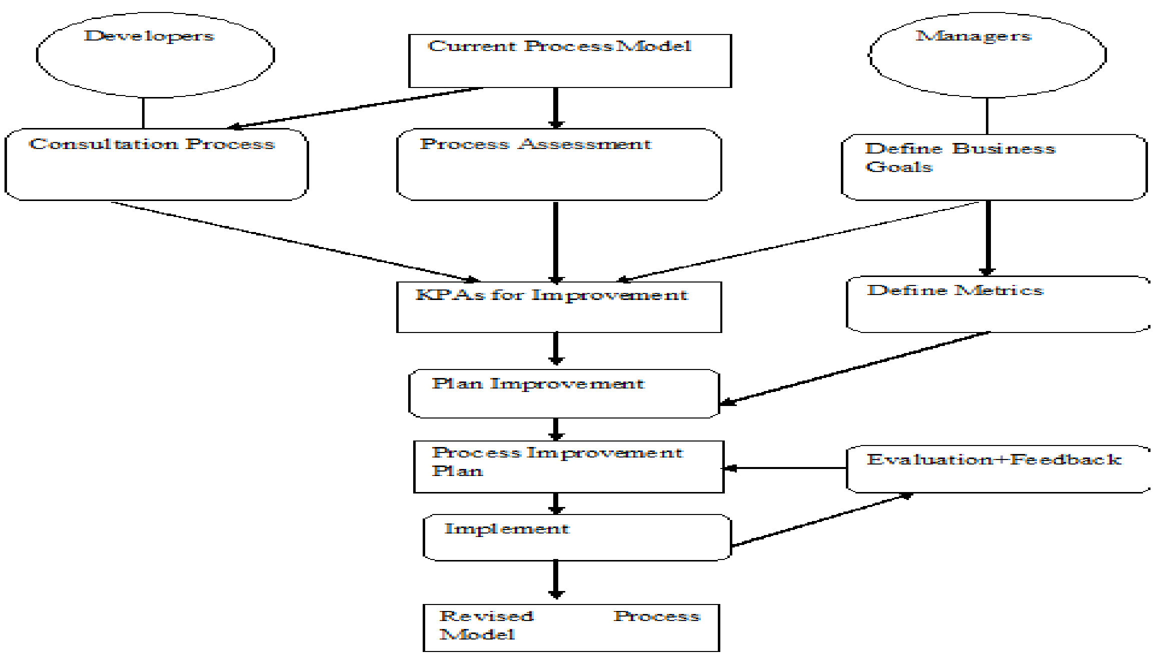

Fig 2.1.19: The PRISMS Process [45]

2.1.22. K-Model

The guideline of software process quality certification consists of project and formation level, and it developed to satisfy the investigation of software process quality capability and improvement at the same time. There are three levels in K-model. They are Initial level (level 1), Good level (level 2) and Very Good level (level 3) [54].

Level 1 is the basic level for improving the capability of the processes in an organization. Level 2 is concerned with developing the project and controlling it and Level 3 is concerned with improving the quality level of the projects and finding out the fundamental reason for improvement through quantitative process management. The K-model is useful to adapt small and medium sized organization. [54]

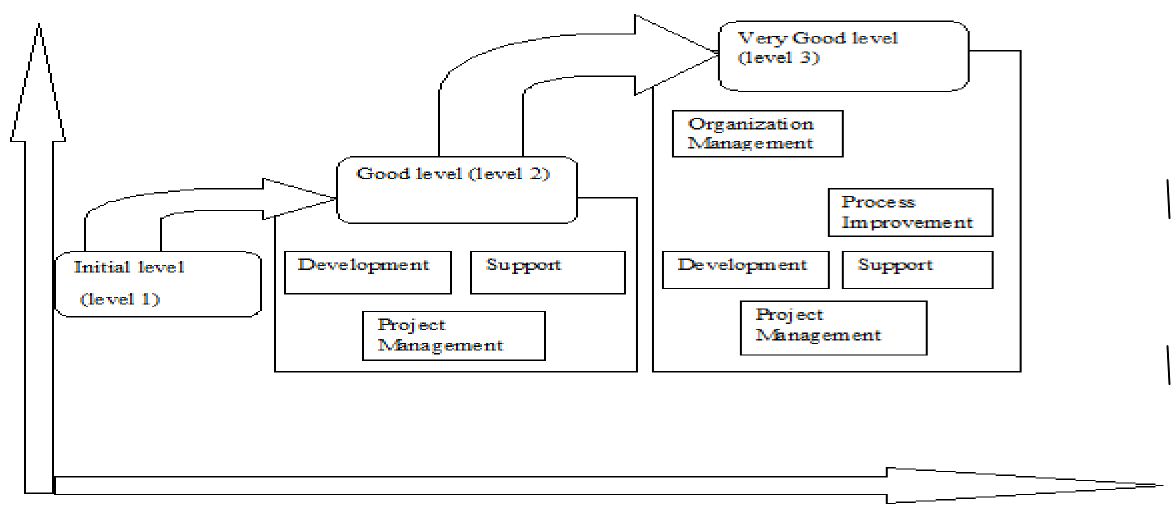

Fig 2.1.20: Steps of process quality certificate in K-model [54]

Level 1 is the basic level for improving the capability of the processes in an organization. Level 2 is concerned with developing the project and controlling it and Level 3 is concerned with improving the quality level of the projects and finding out the fundamental reason for improvement through quantitative process management. The K-model is useful to adapt small and medium sized organization. [54]

2.2 Agile Software Development Methodologies

An iterative and incremental (evolutionary) approach to software development which is performed in a highly collaborative manner by self-organizing teams within an effective governance framework with "just enough" ceremony that produces high quality solutions in a cost effective and timely manner which meets the changing needs of its stakeholders[12][13].

Agility, for software development organization, is the ability to adopt and react expeditiously and appropriately to changes in its environment and to demands imposed by this environment [41] .An agile process is one that readily embraces and supports this degree of adaptability .So, it is not simply about the size of the process or the speed of the delivery; it is mainly about the flexibility.

Eight methodologies of agile software development had been studied which was found effective in the current scenario of software development. They are

1. Extreme Programming (XP)

2. Scrum

3. Crystal Methodologies

4. Unified Process

5. Adaptive Software Development (ASD)

6. Feature Driven Development (FDD)

7. Dynamic Systems Development Method (DSDM)

8. Lean Development

2.2.1. Extreme Programming

The evolution of XP is as a result of the problems caused by the long development cycles of traditional models [22]. XP was initially developed to get the job done [22]. The XP life cycle can be described in five phases. They are Exploration, Planning, Iterations to Release, Productionizing, Maintenance and Death [29].

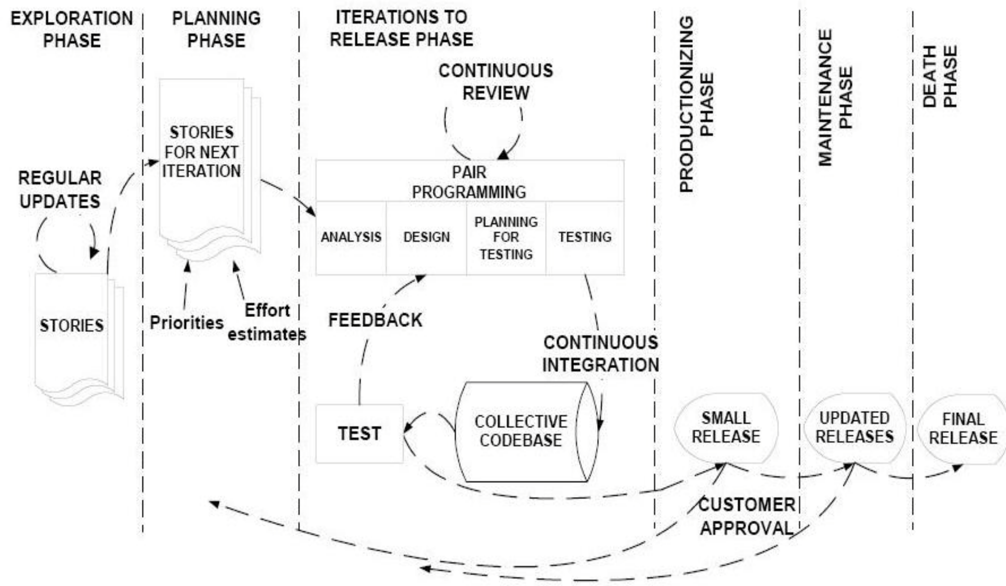

Fig 2.2.1: Life Cycle of the XP process [29]

In exploration phase, the XP team listens to the customers about the requirement of their software and the technical team understands the business context for the software and the required output and major features and functionalities. The requirements of customers lead to creation of user stories which describes the required output, features and functionality for the software that will be built. The customer writes the stories and is placed in the index card. The XP team members assign a value to the stories based on the priority on overall business value of the feature and function. The schedule set in planning stage is broken down to several iterations before the first release.

In productionizing phase, extra testing is done and checking of the performance of the system before it is released to the customer. In maintenance phase, an effort is required for customer support tasks. When the customer has no stories to be implemented, it enters into death phase.

The roles that are described by XP are Programmer, Customer, Tester, Tracker, Coach, Consultant and Manager (Big Boss) [29].

2.2.2. Scrum

Ken Schwaber first described Scrum in 1996 [6] as a process that accepts that the development process is unpredictable, formalizing the “do what it takes” mentality, and has found success with numerous independent software vendors. According to Schwaber [31], Scrum starts with the premise that software development is too complex and unpredictable to be planned exactly in advance. The Scrum approach was developed for managing the systems development process. The industrial process control theory to system development ideas are applied so that flexibility, adaptability and productivity are reintroduced. The Scrum is implemented as an iterative, incremental skeleton.

The Scrum Process Overview is given in the figure below. A high-level vision is developed in a Scrum-based project. A list of known requirements is stored in a Product Backlog list. The customer, sales and marketing division, customer support or software developers can give their requirements. The requirements are prioritized and are divided into small time-box iterations known as Sprints and the effort needed for their implementation is also estimated [31].

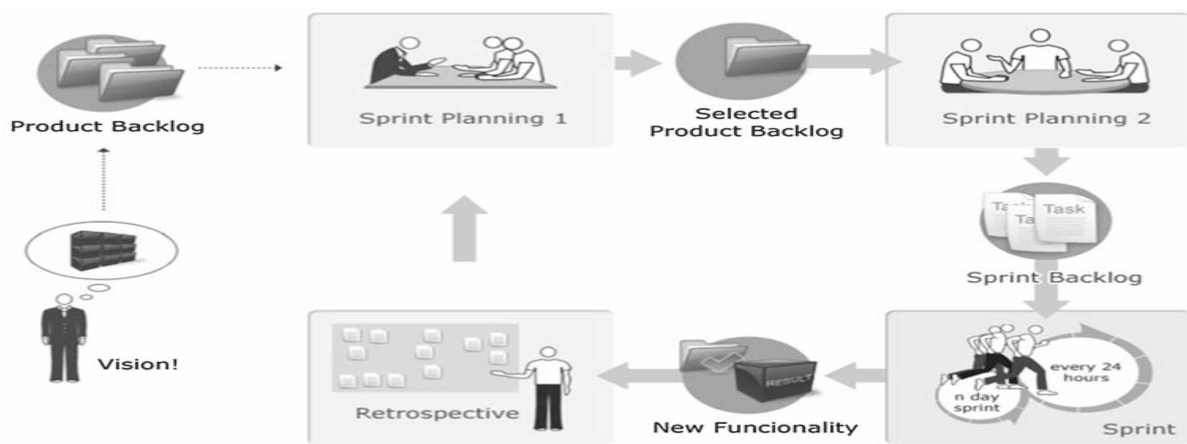

Fig 2.2.2: Scrum Process Overview [31]

The Sprint carries out every task in a Scrum. The development time period in a sprint is 30.The Product Backlog list is constantly updated with new items of requirements. The Scrum team reviews the updated Product Backlog in each iteration. After deciding what has to be done, the Sprint Backlog is developed. During the execution of each Sprint, a daily meeting of 15 minutes are held to monitor work progress and schedule other meetings, if necessary. During everyday Scrum, each team members need to answer three questions in the meeting. They are: What have you done on this project since the last daily Scrum meeting? What will you do before the next meeting? Do you have any obstacles? At the end of each Sprint, the result is presented to the stakeholder by the team [31].

When it is found that the stakeholders had no more requirements, the system is now ready for release and preparation is done which includes task such as integration, system testing and documentation. There are three roles in Scrum. They are Product Owner, Team and Scrum Master [30].

2.2.3. Crystal Family of Methodologies:

The crystal family of methodologies includes family of methodology and is used for selecting the most suitable methodology for each individual project. The Crystal family members are marked with color indicating the ‘heaviness’ of the project. Crystal Clear and Crystal Orange are the two family members that have been constructed and used. The darker the color the more heavier is the methodology. The project size and criticality defines as which methodology color of Crystal is best suited. There are currently three main crystal methodologies constructed: Crystal Clear, Crystal Orange and Crystal Orange Web [2].

Every project which wants to follow Crystal Clear and Crystal Orange need to follow the seven properties of the methodologies. They are: Frequent Delivery, Reflective Improvement, Osmotic Communication, Personal Safety, Focus, Easy Access to Expert users and Technical Environment with Automated tests, Configuration Management and Frequent Integration [6].

In frequent delivery, the developers frequently deliver the usable code to the customers. Reflective Improvement describes the reflective improvement of the processes which defines how can we make this suit us better. The Osmotic communication defines having the entire close to each other so that the people don’t have to go a long way to raise or answer questions. The Personal safety refers to the team members that they feel themselves to be a part of the team and feel able to contribute without fear or reprisal. The focus defines that the members totally concentrates on the task, they have in hand. Easy Access to Expert users defines that if the expert users are easily available to the team, they can answer questions and deliver feedback on quality and design decision. Technical Environment with Automated tests, configuration Management and Frequent Integration defines that a proper technical environment should be present so that testing and configuration management /version control tasks do not have to be done by hand and will make life easier for developers.

In crystal clear, the main roles are sponsor, senior designer programmer, designer programmer and user [3].

2.2.4. Rational Unified Process

The Rational Unified Process or in short RUP is a software design methodology created by the Rational Software Company.RUP is an industry oriented software development methodology which is iterative in nature. It is basically for object oriented system and is developed to complement Unified Modelling Language (UML).The development of software is basically inclined towards object-oriented concept [20].

There are four phases in the life span of a RUP project [42]. They are Inception, Elaboration, Construction and Transition. These phases are split into iterations.

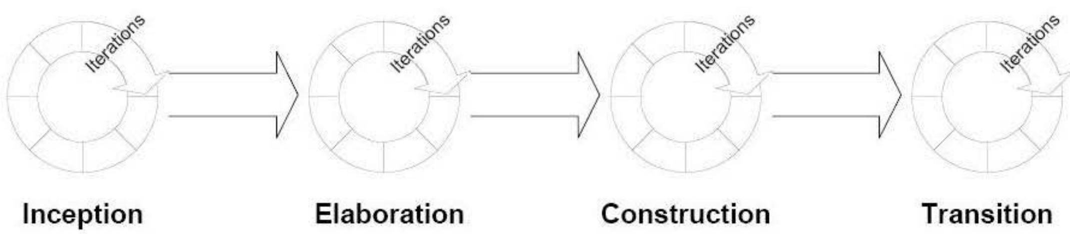

Fig2.2.3: RUP Phases [42]

In Inception phase, every stakeholder for e.g. end user, purchaser, or contractor are considered so that their needs are reflected in the life cycle objectives of the project. Critical use cases are also identified which drives the functionality of the system.

In Elaboration phase, the software architecture is being described out here. The project plan is defined by analyzing the problem domain. The requirements and the plans are assumed to be stable in the elaboration phase. The phase identifies and describes most of the use cases and all users. The prototype of the architecture has been created after the identification of the architecture of the software in this phase. At the end of the phase, it visions the expenditure of the resources with respect to what was planned initially.

In Construction phase, the components that are remained out and the application features are developed, and are integrated to product and are tested. This phase is also considered as manufacturing phase. It emphasizes on managing resources, controlling costs, schedules and quality.

In Transition phase, the product reaches to a stage where the user community is able to use it with no more requirements for the product.

RUP defines 30 roles as workers. They are Architect, Designer, Design Reviewer, Configuration Manager, etc [42].

2.2.5. Adaptive Software Development (ASD)

The Adaptive Software Development (ASD) was developed based on rapid application development. It emphasizes on the continuous adaptation of the process.ASD addresses the economy of increasing returns i.e. the economy characterizes high speed and high change which other traditional approaches are unable to handle it.ASD methodology encourages incremental, iterative development with constant prototyping [23].

There are three-phase cycles in ASD [23]. They are Speculate, Collaborate and Learn. They are shown in the fig below. The Project Initiation phase defines the objectives and goal of the project. The overall schedule, as well as the schedules and objectives of the development cycle are fixed in the Project Initiation phase. The ASD is component-oriented i.e. they focus on the results and quality. The Collaborate phase undergoes adaptive development cycles where several components undergo development concurrently. The quality review is done with the presence of the customer as a group of experts which focuses on the functionality of the software developed during the cycle. The final stage of ASD is the Final Q/A and Release stage. The sequence of the phases are not so important in ASD what’s important is the learning of the lessons.

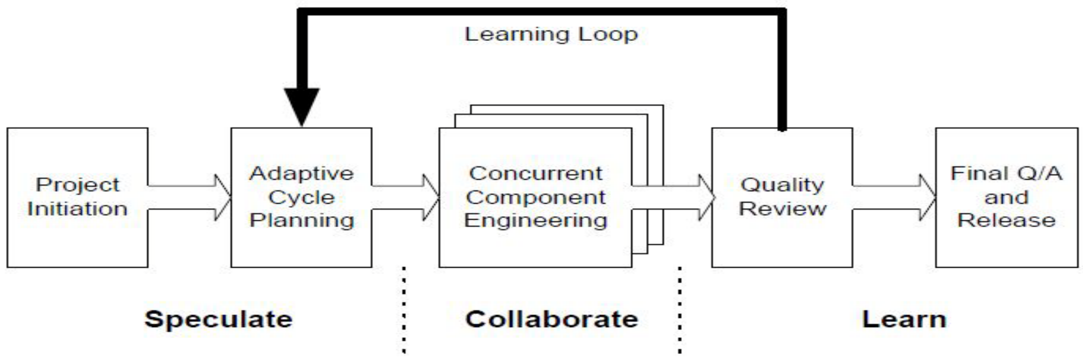

Fig 2.2.4: The ASD Lifecycle Phases [25]

The roles of ASD are Executive Sponsor, a facilitator to plan and lead the session, a scribe to take the minutes, the project manager, and customer and developer representative. [23]

2.2.6. Feature Driven Development (FDD)

Feature Driven Development is an agile and adaptive process that focuses on the design and building phases. The FDD approach is designed to work with any software development project .The approach doesn't require any specific process model and follows iterative development [51].

The FDD approach consists of five sequential processes. The designing and building is carried out in these processes which are iterative in nature and have quick adaptations to late changes in requirements and business needs [51].

The first phase known as Develop and Overall Model, the domain experts begins developing the overall model, where the scope, contexts and requirements of the system are already known to the domain experts. Different domain areas are being created from the overall domain so that the domain members had a detailed walkthrough for each of the domain areas.

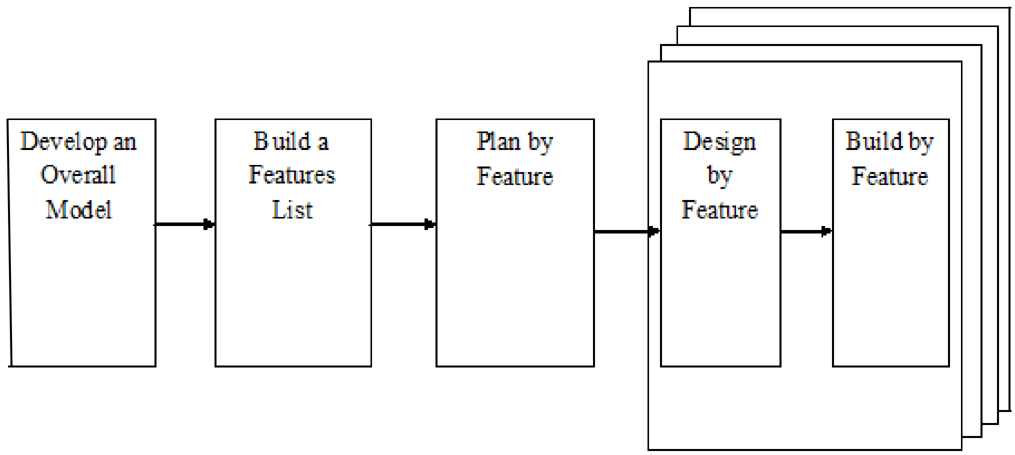

Fig2.2.5: Processes of FDD [51]

In Build a Features List, a comprehensive feature list is being built for the system that is to be developed from the walkthroughs, objects models and existing requirement documentation. The sub feature sets are created from the overall feature set which represent different activities within specific domain areas.

In Plan by feature, the features are planned out so that a high-level plan is created with the priority and dependencies of feature sets and is assigned to Chief Programmers.

In Design by Feature and Build by Feature, the Class Owners selects a small group of features from the feature set(s) and feature teams in order to develop selected features. The design by feature and build by feature are iterative when the selected features are produced. Multiple feature teams are designed and built concurrently. The features that are completed after successful iterations are promoted to the main build. The iterations again begins with the designing and building of new group of features are taken from the feature set.

The roles that are present in FDD are Project Manager, Chief Architect, Development Manager, Chief Programmer, Class Owner, Domain Expert, Domain Manager, Release Manager, Language Lawyer/Language GuruBuild Engineer, Toolsmith, System Administrator, Tester, Deployer and Technical Writer [51].

2.2.7. Dynamic Systems Development Method (DSDM)

DSDM is a framework for RAD development which is non-profit and non-proprietary. The main concept behind DSDM approach is that it prefers to fix time and resources and then accordingly adjust the functionality [26].

DSDM consists of five phases [26]. They are feasibility study, business study, functional model iteration, design and build iteration, and implementation.

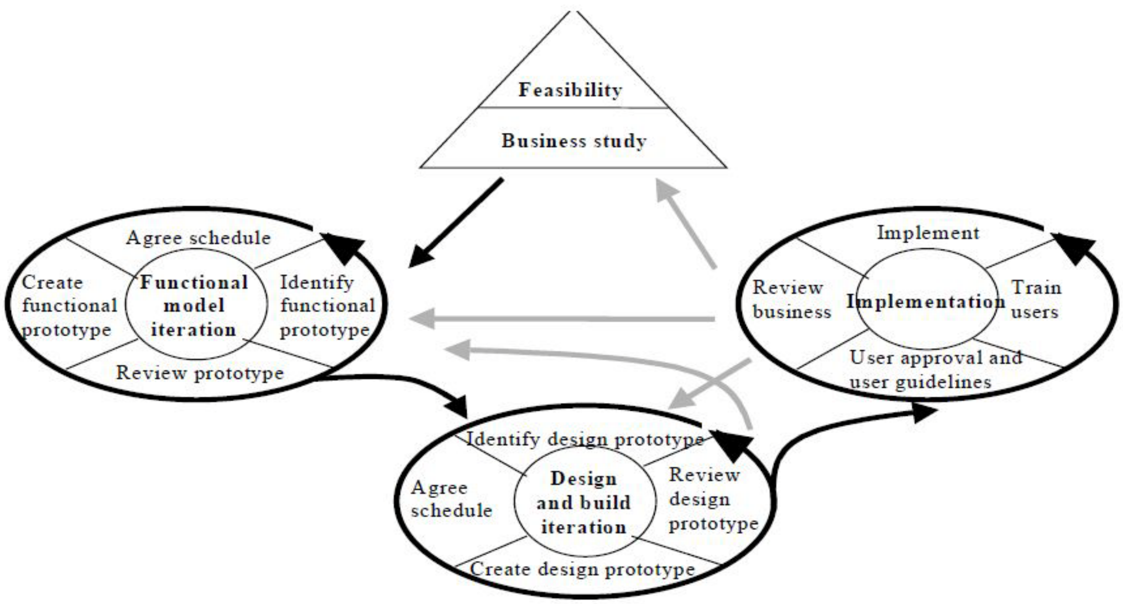

Fig 2.2.6: The DSDM Process Diagram

In feasibility study, the project is assessed for the suitability of DSDM. In business study phase, the analysis of essential characteristics of business and technology is done. The first iterative and incremental phase is the functional model iteration phase. At first the functional prototype is created and the developers gains experiences from them and are used in improving the analysis model. The output results in a functional model which contains the prototype code and the analysis model. In this phase, testing is also a very mandatory part. At different stages in the phase, there are further four outputs. They are Prioritized functions, Functional prototyping review documents, on-functional requirements and Risk Analysis of further development. In design and build iteration phase, the system is built in this phase. In final implementation phase, the system is shifted to the production environment.

The DSDM have Developers and Senior Developers, Technical Coordinator, Ambassador User, Adviser User, Visionary and Executive Sponsor as roles in the approach [26].

2.2.8. Lean Software Development: The objective of Lean Software Development is to achieve breakthrough quality, savings, speed, and business values in a project. There are seven lean principles that have revolutionized the manufacturing and research and development(R & D).The seven principles are described as follows [39].

Elimination of waste defines anything that doesn’t add value to a product i.e. value as perceived by the customer. Amplify learning helps software development teams to improve the software development environment. When the domains involve uncertainty, it is better to take decisions as late as possible because decisions that are made by delaying are based on fact and not on speculation. Deliver as fast as possible speed assures what customer needs today and not what they wanted yesterday. Empower the team by involving the people i.e. the developers who is actually doing the work inorder to achieve excellency in work. Building integrity in a system defines that a user gets from the system what he/she actually wants and to get integrity for products in the market depends on customer perception overtime. It is necessary to focus equally on different diverse areas of the system so that it results the overall system performance [39].

Chapter 3
Problem Statement

A software organization, whether small , medium or large scale wish to succeed on the market by providing software of high quality together with related services, support, communication with customers etc. The small and medium scale enterprises (SMEs) are one of the most developing sectors that reflect the economy of a nation. Now a days, this very promising sector is striving for standardization of their software development processes. They are carrying out improvement processes but not a proper set of processes. And also, there is limited adoption, absorption of software process improvement models in SME’s due to lack of know-how and available resources in terms of money, time, perceived benefits, quality focus. The company’s confidence is gained by increasingly the quality of the software and maturing the software approaches.

Software Process Improvement models such as CMM, P-CMM, SPICE, are implemented to improve quality of the product. But these models are applicable in large scale enterprises but don’t fit in small and medium scale enterprises (SMEs). Because the SMEs are unable to bear the cost of these models in implementing software process improvement programs and also because of limited resource and strict time constraints. The models are basically written keeping in mind the large scale enterprises.

It is seen that the SMEs failed to improve their software product quality because it is largely dependent on the process that is used to create it. The main drawback of the small and medium scale enterprises (SMEs) is that they do not have a structured process culture in their organization. Therefore, customer dissatisfaction, inadequate software quality, inability to deliver on time and within budget and excessive rework are some of the byproducts of their haphazard approach. So, the main objective of thesis is to propose a Software Process Improvement Model which is easily incorporated and adopted by small and medium scale enterprises (SMEs).

Chapter 4
Analysis of the Conducted Survey in SMEs

4.1 Organization of Survey

A literature survey of 22 software process improvement models and eight Agile Methodologies is studied. These SPI Models are Capability Maturity Model (CMM), Personal Software Process Model, Team Software Process (TSP), Capability Maturity Model Integration (CMMI),Six-Sigma (SS), People-CMM (P-CMM),Malcolm Balridge National Quality Award (MBNQA),System Engineering CMM (SE-CMM), System Security Engineering CMM (SSE-CMM),ISO 9000,Trillium Model, SPICE, ISO/IEC 12207, BOOTSTRAP,SECAM, SDCE,MIL-STD-498,IEEE 1220, TickIT, IDEAL Model, PRISMS and K-Model. The eight agile methodologies that are studied are Extreme Programming, Scrum, Crystal Methodologies, Rational Unified Process, Adaptive Software Development, Feature Driven Development, Dynamic Systems Development Method and Lean Development The process areas from different SPI models are studied and the commonalities of different process areas are grouped and compared with the activities of different Agile Methodologies.

A survey questionnaire consisting of 150 questions is made by taking into account the process areas that are applicable in Small and Medium Enterprises(SMEs) which can be easily affordable by the SMEs with less cost and those produce quality products in time. The survey is carried out in 18 SMEs of India.They are FutureSoft, Delhi, IT Pyramid, Noida, Zaloni Technologies, Guwahati, DZ Engineering, Pune, Aon Hewitt, Chennai, CGI Technologies, Bangalore, IAG Automation, Gurgaon, SIQUES, Noida, Vacpic Technologies Pvt Ltd, Banagalore, Aris Global Software Pvt Ltd, Yodlee Pvt Ltd, Roma Think Soft, Bangalore, Xixom Technology, Delhi, IQ Systems India Pvt Ltd, Chennai, Globrin Technologies, Gurgaon, Exilant Technologies Pvt Ltd, Bangalore, and Verchaska Infotech Pvt Ltd, Mumbai. The questionnaires are being filled up by Senior Managers, Project Managers, Team Leader and the Software Engineers.

4.2 Analysis and Conclusions

i) It has been seen that the SMEs are using to a great extent the agile methodologies in order to improve their productivity and quality.

ii) Scrum is favourite for most of SMEs in India. And these organizations are doing well in Software Process Improvement (SPI) also with Scrum instead of limited resources. Only the different Scrum Team Requirement ask for more human resources in SMEs which they are managing by making smaller teams i.e. they make all different teams needed but allotting less number of persons per team. Concluding, small and medium software organizations have found a superb Agile Methodology as Scrum.

iii) The percentage of use of Agile Methodologies in different Indian SMEs are given in the Figure 4.1 below:

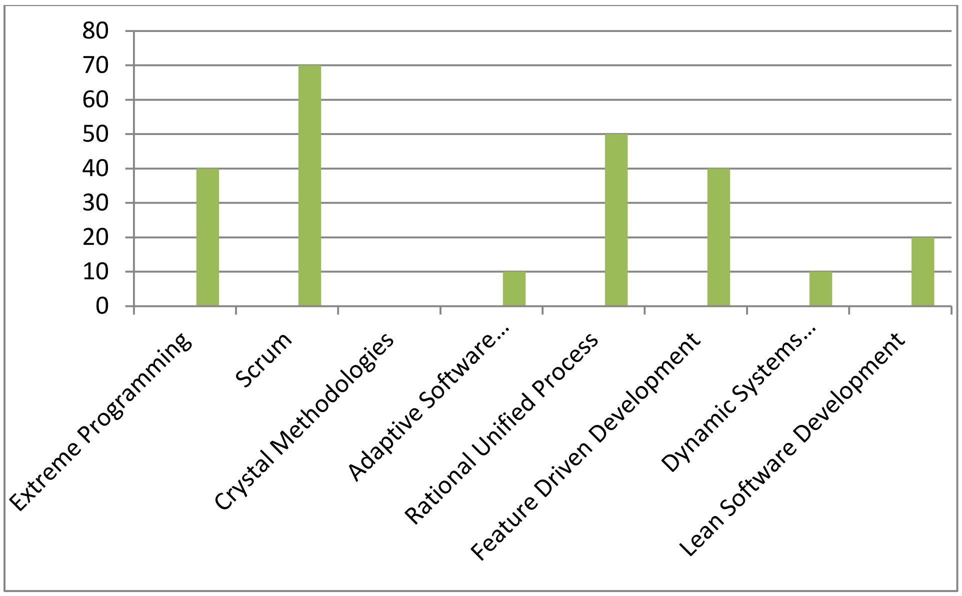

Fig 4.1: Percentage of Use of Agile Methodologies in SMEs of India

It has been seen that Scrum is favourite for most of SMEs in India. And these organizations are doing well in Software Process Improvement (SPI) also with Scrum instead of limited resources. Only the different Scrum Team Requirement ask for more human resources in SMEs which they are managing by making smaller teams i.e. they make all different teams needed but allotting less number of persons per team.

iv) It has also been seen that though the SMEs are not using any proper SPI models, they are using different Software Development Tools in their organization to improve their productivity and quality. The different Software Developments tools are Requirement Management Tools, Software Project Planning Tools, Process Modelling Tools, Software

Configuration Management Tools and Cost Models. The different figures below show the percentage of use different Software Development Tools in Indian SMEs.

a). Percentage of Use of Requirement Management Tools in SMEs of India:

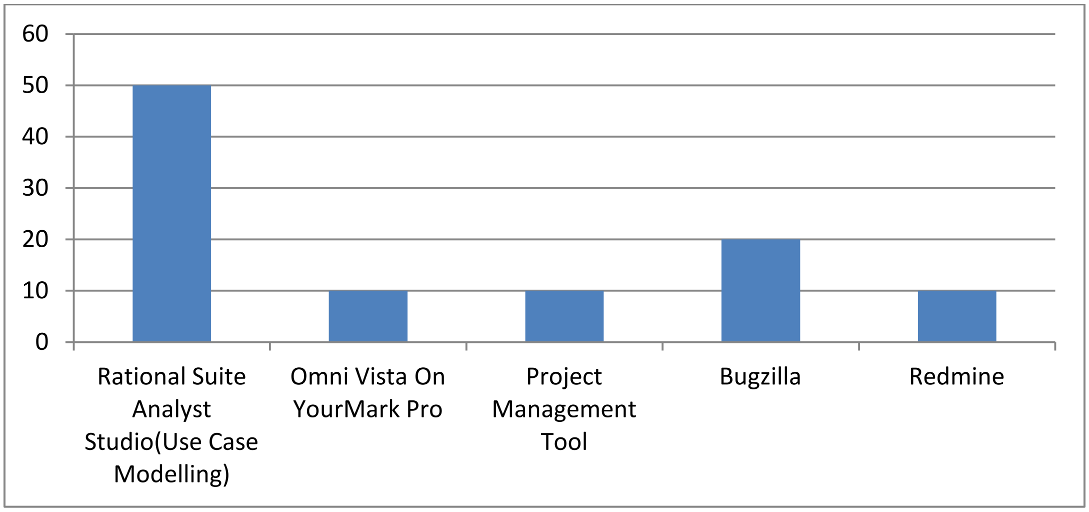

Fig 4.2: Percentage of Use of Requirement Management Tools in SMEs of India

It has been seen in Fig 4.2 most of the Indian SMEs uses Rational Suite Analyst Studio (Use Case Modelling) as requirement management tool in their organization while the other requirement tools are being used by different SMEs on small scale.

b).Percentage of Use of Software Project Planning Tools in SMEs of India

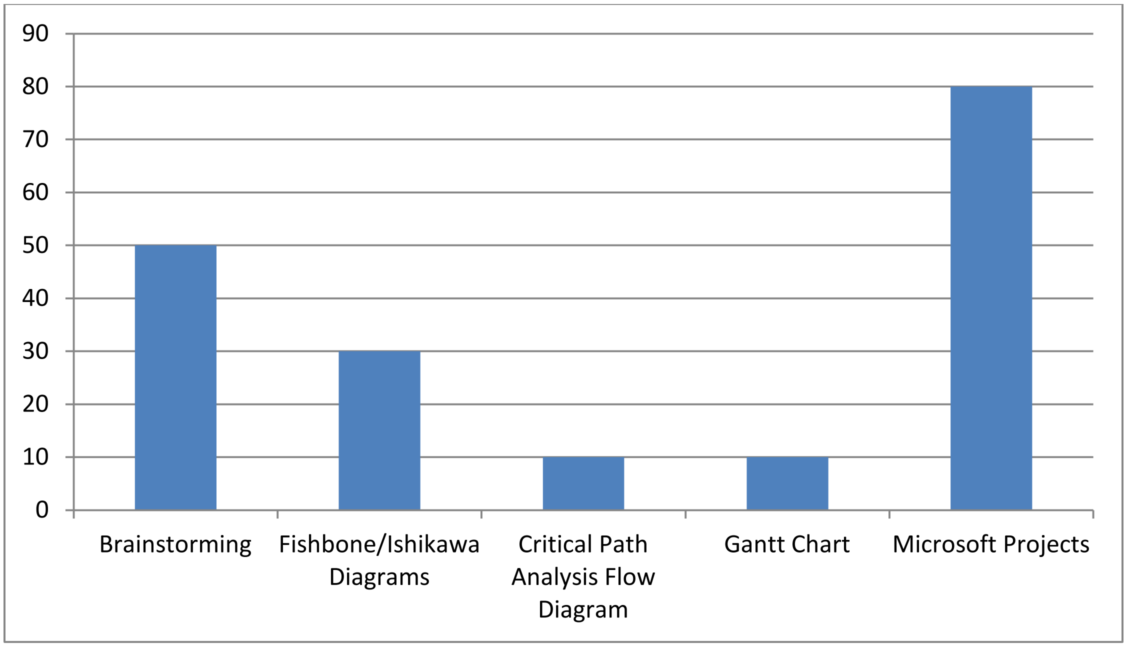

Fig 4.3: Percentage of Use of Software Project Planning Tools in SMEs of India

It has been seen from Fig 4.3 that most of the Indian SMEs use Microsoft Projects as Software Project Planning Tools, while the other Software Project Planning Tools are being used by some SMEs of India.

C. Percentage of Use of Process Modeling Tools in SMEs of India

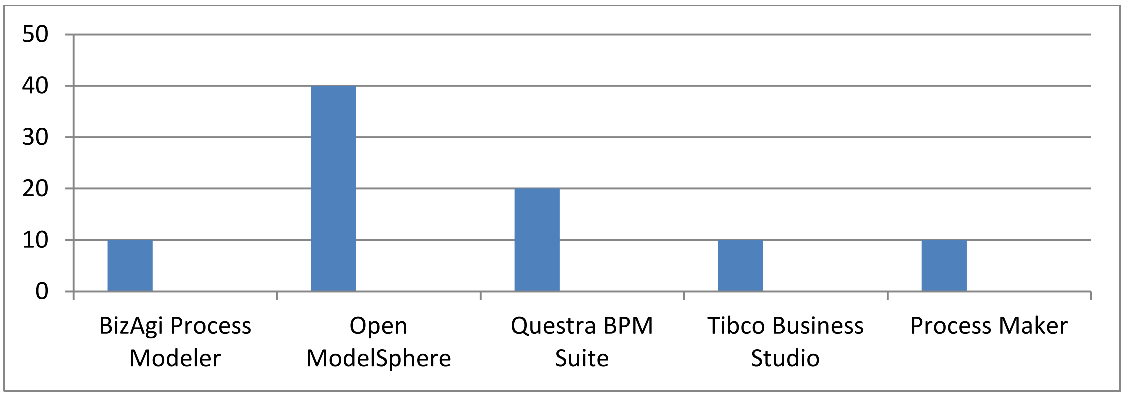

Fig 4.4: Percentage of Use of Process Modeling Tools in SMEs of India

It has been seen from Fig 4.4 that most of the Indian SMEs uses Open ModelSphere as Software Project Modeling Tool, while the other Process Modeling Tools are liked by few SMEs of India.

D. Percentage of Use of Software Configuration Management Tools in SMEs of India

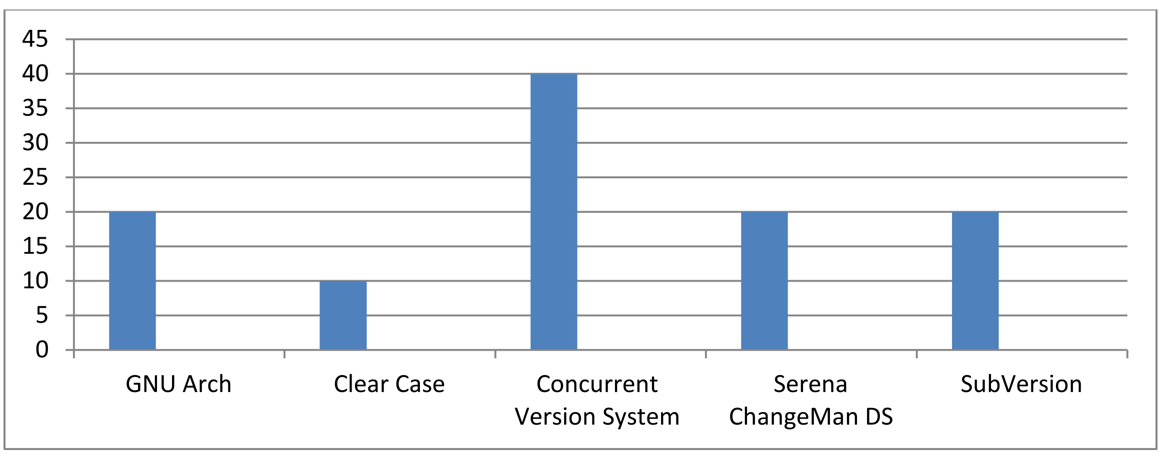

Fig 4.5: Percentage of Use of Software Configuration Management Tools in SMEs of India

It has been seen in Fig 4.5 that most of the Indian SMEs uses Concurrent Version System as Software Configuration Management Tools, while the other Software Configuration Management Tools like GNU Arch, Clear Case etc are being used by different SMEs of India.

E. Percentage of Use of Cost Estimation Models in SMEs of India

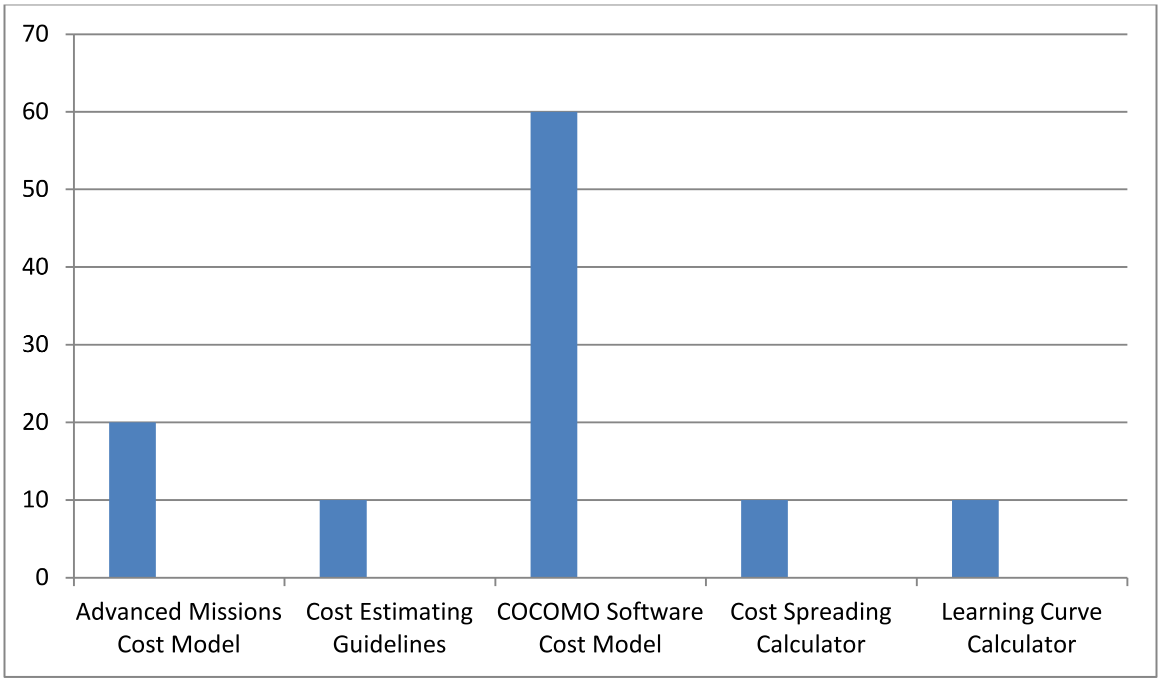

Fig 4.6: Percentage of Use of Cost Models in SMEs of India

It has been seen in Fig 4.6 that most of the Indian SMEs use COCOMO Software Cost Model as Cost Estimation Models, while the other Cost Models are being used randomly by different SMEs of India.

Chapter 5
SPI Expert in SMEs

5.1 SPI Expert in SMEs

SPI Expert in SMEs is the name given to the proposed model. Based on the findings of the survey, SPI expert concentrates on the organization's choice of processes those are given in support set. SPI expert contains SPI Expert Support Set which inherently is two staged model: The Basic Stage and the Elaborated Stage. These stages are actually materialized in given support set. And the SPI Expert Support Set is embedded in Process Flow.

The main highlight of the model is SPI Expert Support Set, from which the user of the model can select the processes according to the organization's budget, time and product's quality constraints. If only basic level quality to be maintained, processes in basic stage are needed to be incorporated. Otherwise, user of the model can select from the Elaborated Stage. If elaborated Stage's processes needed to be selected, then by default basic stage's processes become mandatory to be carried on. Some mandatory and optional processes are quoted in Elaborated stage. Mandatory one needs to be selected surely if user selects to be in elaborated stage. The optional process depicts more effective and enhanced process which can improvise product quality. Optional processes can incur more cost/time also.

Referring Fig 5.1, SPI Expert in SMEs illustrated below:

i) The Process Flow depicts The Product Backlog which contains the requirements as asked by the customer for a particular project. The requirements are divided into requirements set as requirement set 1, requirement set 2 ………... requirement set n in accordance with priority. The requirement set in the Product Backlog with highest priority is taken out by the experts to be discussed in the Process Kick Off Expert Meeting.

ii) In the meeting, the phases to be incorporated are discussed by the experts. This decision is taken based on the type of project in hand. The schedule of the phases is decided in the meeting.

iii) The phases are mapped onto the support set which contains the different process areas. The support sets contains sets of process areas. The sets of process areas are Requirement Management, Planning, Testing, Risk Management, Measurement and Analysis and

Reuse(Optional).The different process support sets have their unique ids and also the process areas under each support set have unique ids which shows to which support set do they belong. For different Process Kick Off Expert Meeting, the schedule of the phases can be different.

iv) After completing the phases chosen from the support set, the project undergoes through the Quality Control.

v) It is discussed with the customers after Quality Control. If the customers agree to it, the product is completed and delivered.

vi) It is proceeded further to check whether any new process area needed to be incorporated in support set. If required it is added and tabulated in process database only on the affirmation by experts. The experts also decides under which category of support set the process area falls and is then added accordingly in SPI Expert's process database .

vii) If after Quality Control the customers are not satisfied, Process Kick Off Expert Meeting is held and the scenario is repeated once again.

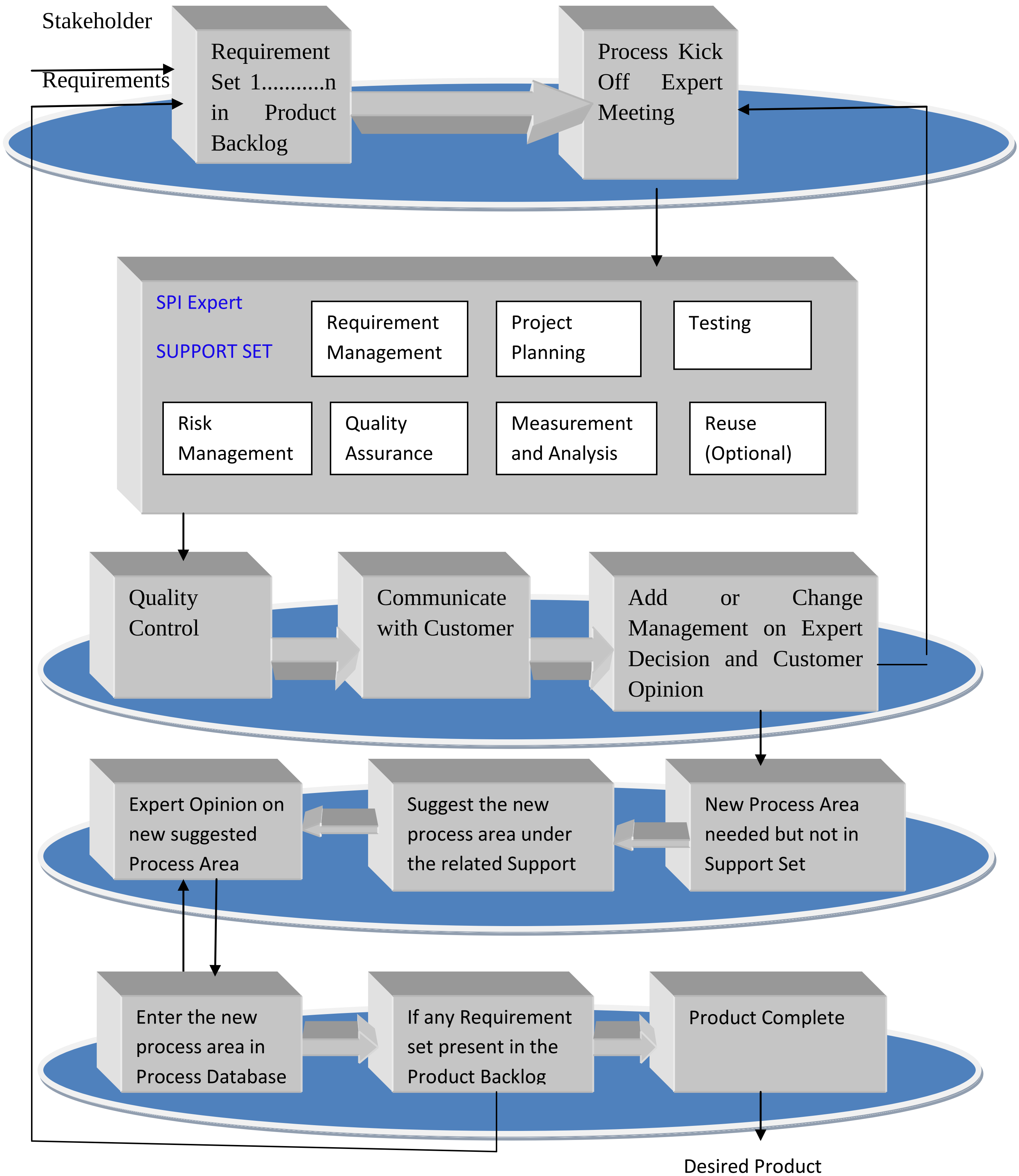

Fig 5.1: SPI Expert in SMEs

5.2 SPI Expert Support Set

The SPI Expert support set contains sets of process areas. The sets of process areas are Requirement Management, Planning, Testing, Risk Management, Measurement and Analysis and Reuse(Optional). All Process areas are tabulated in Table 1, Table 2………Table 8. The different process support sets have their unique ids and also the process areas under each support set have a unique id which shows to which support set do they belong. For different Process Kick Off Expert Meeting, the schedule of the phases can be different. SPI Expert Support Set contains two stages: The Basic Stage and the Elaborated Stage. The user of the model can select the processes according to the organization's budget, time and product's quality constraints. If only basic level quality to be maintained, processes in basic stage are needed to be incorporated. Otherwise, user of the model can select from the Elaborated Stage. If elaborated Stage's processes needed to be selected, then by default basic stage's processes become mandatory to be carried on. Some mandatory processes (specified by M) and optional processes (specified by M) are quoted in Elaborated stage. Mandatory one needs to be selected surely if user selects to be in elaborated stage. The optional process depicts more effective and enhanced process which can improvise product quality.

i) Requirement Management Process Areas (REQM)

REQM 1.	Basic Level	REQM 2.	Elaborated Level	Documentation
REQM 1.1.	Requirement Gathering	REQM 2.1.	Understanding between customer and developer(M)	Scope of the project, Requirements given by Customer.
		REQM 2.1.1.	Requirement Reuse(O)	Scope of the project, Requirements given by Customer.
		REQM 2.1.2.	Safety Requirements (O)	Scope of the project, Requirements given by Customer.
REQM 1.2.	Requirement Analysis	REQM 2.2.	Identify Requirement Change (M)	Scope of the project, Requirements given by Customer.
REQM 1.3.	Identification of Customer Needs	REQM 2.3.	Bi-directional traceability of requirements (M)	Scope of the project, Requirements given by Customer.
		REQM 2.4.	Documentation of requirements and change	Scope of the project, Requirements given by

			requirements (M)	Customer.
		REQM 2.5.	Requirements Validation(M)	Scope of the project, Requirements given by Customer.
		REQM 2.6.	Requirements Verification(M)	Scope of the project, Requirements given by Customer.

Table 1: Requirement Management Process Areas(REQM)

ii) Planning Process Areas(PLAN)

PLAN 1.	**Basic Level**	**PLAN 2.**	**Elaborated Level**	**Documentation**
PLAN 1.1.	Scope of the project	PLAN 2.1.	Estimation of effort and cost (M)	Cost Benefit Analysis Documernt, Risk Document,Project Plan
		PLAN 2.1.1	Stakeholder Involvement (O)	Scope of the project
		PLAN 2.1.2	Reconcile Work and Resource Levels (O)	Cost Benefit Analysis Documernt, Risk Document,Project Plan
		PLAN 2.1.3	Training of Developers and Users (O)	Cost Benefit Analysis Documernt, Risk Document, Project Plan
		PLAN 2.1.4	Technology Management (M)	Cost Benefit Analysis Documernt, Risk Document, Project Monitoring the project, Meeting Minutes
PLAN 1.2.	Define the project life cycle	PLAN 2.2.	Data Management(M)	Results of Monitoring the project, Meeting Minutes
PLAN 1.3.	Project Risks	PLAN 2.3.	Organization Standards and Procedures (M)	Risk Assessment Metrics(RAM)
		PLAN 2.3.1.	Establishing Software Development Environment and Infrastructure (O)	Cost Benefit Analysis Documernt, Risk Document,Project Plan
PLAN 1.4.	Establish Project Plans	PLAN 2.4.	Project Resources(M)	Cost Benefit Analysis Documernt, Risk Document,Project Plan
PLAN 1.5.	Allocation of roles to project	PLAN 2.5.	Review Project Plans(M)	Cost Benefit Analysis Documernt, Risk

	members			Document,Project Plan
		PLAN 2.6.	Plan Commitments(M)	Cost Benefit Analysis Documernt, Risk Document,Project Plan
		PLAN 2.7.	Documentation and Record Management(M)	Cost Benefit Analysis Documernt, Risk Document,Project Plan
		PLAN 2.8.	Software Architecture Design and Review Process(M)	Cost Benefit Analysis Documernt, Risk Document,Project Plan
		PLAN 2.9	Progress Review (M)	Cost Benefit Analysis Documernt, Risk Document,Project Plan

Table2: Planning Process Areas(PLAN)

iii) Testing Process Areas(TEST)

TEST 1.	Basic Level	TEST 2.	Elaborated Level	Documentation
TEST 1.1.	Test Cases	TEST 2.1.	Requirements Testing (M)	Master Test Plan, Test Cases, Test Procedures, Test Logs, Test Data, Test Summary Report
TEST 1.2.	Software Coding and Unit Testing	TEST 2.2.	Performance Testing(M)	Master Test Plan, Test Cases, Test Procedures, Test Logs, Test Data, Test Summary Report
TEST 1.3.	Integration Testing	TEST 2.3.	Regression Testing(M)	Master Test Plan, Test Cases, Test Procedures, Test Logs, Test Data, Test Summary Report
TEST 1.4	System Testing	TEST 2.4.	Manual Support Testing(M)	Master Test Plan, Test Cases, Test Procedures, Test Logs, Test Data, Test Summary Report
TEST 1.5	System Integration Testing	TEST 2.5.	Acceptance Testing(M)	Master Test Plan, Test Cases, Test Procedures, Test Logs, Test Data, Test Summary Report
		TEST 2.6.	Build Test Teams(O)	Master Test Plan
		TEST 2.7.	Test Planning and Estimation(M)	Master Test Plan, Test Cases, Test Procedures, Test Logs, Test Data.
		TEST 2.8	Test Results Reporting (O)	Test Summary Report

		TEST 2.9	α test and β test (M)	Master Test Plan, Test Cases, Test Procedures, Test Logs, Test Data, Test Summary Report
		TEST 2.10	Test Execution and Defect Tracking (O)	Master Test Plan, Test Cases, Test Procedures, Test Logs, Test Data, Test Summary Report
		TEST 2.11	Test Reports and Acceptance (O)	Master Test Plan, Test Cases, Test Procedures, Test Logs, Test Data, Test Summary Report.
		TEST 2.12.	Test Metrics (O)	Master Test Plan, Test Cases, Test Procedures, Test Logs, Test Data.
		TEST 2.12.i.	Number of test cases planned (M)	Master Test Plan, Test Cases, Test Procedures, Test Logs, Test Data.
		TEST 2.12.ii.	Number of test cases failed (M)	Master Test Plan, Test Cases, Test Procedures, Test Logs, Test Data.
		TEST 2.12.iii.	Number of test cases passed (M)	Master Test Plan, Test Cases, Test Procedures, Test Logs, Test Data.
		TEST 2.12.iv.	Number of bugs opened (M)	Master Test Plan, Test Cases, Test Procedures, Test Logs, Test Data.
		TEST 2.12.v.	Number of bugs closed per unit of time (M)	Master Test Plan, Test Cases, Test Procedures, Test Logs, Test Data.
		TEST 2.12.vi.	Testing changes vs release changes (O)	Master Test Plan, Test Cases, Test Procedures, Test Logs, Test Data.
		TEST 2.12.vii.	Test Effectiveness ratio (M)	Master Test Plan, Test Cases, Test Procedures, Test Logs, Test Data.
		TEST 2.13.	Test Environments (O)	Master Test Plan, Test Cases, Test Procedures, Test Logs, Test Data.
		TEST 2.14	Test Process Improvement (M)	Master Test Plan, Test Cases, Test Procedures, Test Logs, Test Data, Test Summary Report.
		TEST 2.14.i.	Analytical risk-based testing strategies (O)	Master Test Plan, Test Cases, Test Procedures, Test Logs, Test Data, Test Summary

				Report.
		TEST 2.14.ii.	Realistic objectives for testing (M)	Master Test Plan, Test Cases, Test Procedures, Test Logs, Test Data, Test Summary Report.
		TEST 2.14.iii.	Continuous Test Process Improvement (M)	Master Test Plan, Test Cases, Test Procedures, Test Logs, Test Data, Test Summary Report.
		TEST 2.14.iv.	Trained and Certified Test Teams (O)	Master Test Plan
		TEST 2.14.v.	Distribute Testing work intelligently (M)	Master Test Plan
		TEST 2.15	Test System Design and Implementation (M)	Master Test Plan, Test Cases, Test Procedures, Test Logs, Test Data.
		TEST 2.16	Test Automation(M)	Master Test Plan, Test Cases, Test Procedures, Test Logs, Test Data,

Table 3: Testing Process Areas (TEST)

iv) Quality Assurance Process Areas(QAS)

QAS 1.	**Basic Level**	**QAS 2.**	**Elaborated Level**	**Documentation**
QAS 1.1.	Measuring Quality	QAS 2.1.	Standard Procedures, Techniques and Tools (M)	Schedule of Tasks and Responsibilities, System Documentation, Documentation of Standard Practices, Reviews and audits, Problem Reporting and Corrective Action.
QAS 1.2.	Reducing Project & Product Risk	QAS 2.1.i.	Process Quality Assurance(Measurement and Feedback) (M)	Schedule of Tasks and Responsibilities, System Documentation, Documentation of Standard Practices, Reviews and audits, Problem Reporting and Corrective Action.
		QAS 2.1.ii.	Quality Assurance Team (M)	Schedule of Tasks and Responsibilities, System Documentation, Documentation of Standard Practices, Reviews and audits, Problem Reporting

				and Corrective Action.
		QAS 2.1.iii.	Continually Improve Processes (M)	Schedule of Tasks and Responsibilities, System Documentation, Documentation of Standard Practices, Reviews and audits, Problem Reporting and Corrective Action.
		QAS 2.2.	Software Configuration Management (M)	Schedule of Tasks and Responsibilities, System Documentation, Documentation of Standard Practices, Reviews and audits, Problem Reporting and Corrective Action.
		QAS 2.2.i.	Configuration Identification (M)	Schedule of Tasks and Responsibilities, System Documentation, Documentation of Standard Practices, Reviews and audits, Problem Reporting and Corrective Action.
		QAS 2.2.ii.	Configuration Control (M)	Schedule of Tasks and Responsibilities, System Documentation, Documentation of Standard Practices, Reviews and audits, Problem Reporting and Corrective Action.
		QAS 2.2.iii.	Configuration Status Accounting (O)	Schedule of Tasks and Responsibilities, System Documentation, Documentation of Standard Practices, Reviews and audits, Problem Reporting and Corrective Action.
		QAS 2.2.iv.	Configuration Auditing (O)	Schedule of Tasks and Responsibilities, System Documentation, Documentation of Standard Practices, Reviews and audits, Problem Reporting and Corrective Action.
		QAS 2.3.	Change Control (O)	Schedule of Tasks and Responsibilities, System Documentation, Documentation of Standard Practices, Reviews and audits, Problem Reporting and Corrective Action.

		QAS 2.4.	Formal Technical Reviews (M)	Schedule of Tasks and Responsibilities, System Documentation, Documentation of Standard Practices, Reviews and audits, Problem Reporting and Corrective Action.
		QAS 2.5.	Statistical Quality Assurance (M)	Schedule of Tasks and Responsibilities, System Documentation, Documentation of Standard Practices, Reviews and audits, Problem Reporting and Corrective Action.
		QAS 2.6.	Process Management (M)	Schedule of Tasks and Responsibilities, System Documentation, Documentation of Standard Practices, Reviews and audits, Problem Reporting and Corrective Action.
		QAS 2.6.i.	Clarify current knowledge of the process (O)	Schedule of Tasks and Responsibilities, System Documentation, Documentation of Standard Practices, Reviews and audits, Problem Reporting and Corrective Action.
		QAS 2.6.ii.	Identify the process needed to be improved (O)	Schedule of Tasks and Responsibilities, System Documentation, Documentation of Standard Practices, Reviews and audits, Problem Reporting and Corrective Action.

Table 4: Quality Assurance Process Areas(QAS)

v) Risk Management Process Areas(RIMA)

RIMA 1.	Basic Level	RIMA 2.	Elaborated Level	Documentation
RIMA 1.1.	Identification of Risk	RIMA 2.1.	Planning of risk (M)	Categorization of Risks,Prioritization of Risk,Risk Impact Assessment
RIMA 1.2.	Risk Analysis	RIMA 2.2.	Tracking of Risk(M)	Categorization of Risks,Prioritization of Risk,Risk Impact Assessment
		RIMA 2.2.	Controlling of Risks(M)	Categorization of Risks,Prioritization of Risk,Risk Impact Assessment
		RIMA	Risk	Categorization of

		2.2.	Communication(M)	Risks,Prioritization of Risk,Risk Impact Assessment

Table 5: Risk Management Process Areas(RIMA)

vi) Measurement and Analysis Process Areas(MAA)

MAA 1.	Basic Level	MAA 2.	Elaborated Level	Documentation
MAA 1.1.	Target Response Time	MAA 2.1.	Project Metrics (M)	Categorization of Metrics, Results after Measurement,Comparision of the results in accordance to planning.
MAA 1.2.	Length of Source Code	MAA 2.1.i.	Size of the Project(M)	Categorization of Metrics,Results after measurement,Comparision of the results in accordance to planning.
MAA 1.3.	Duration of Testing Process	MAA 2.1.ii.	Cost of the Project(M)	Categorization of Metrics,Results after measurement,Comparision of the results in accordance to planning.
MAA 1.4.	Time a Programmer spends on a project	MAA 2.1.iii.	Time over/under run (M)	Categorization of Metrics,Results after measurement,Comparision of the results in accordance to planning.
MAA 1.5.	Usability	MAA 2.1.iv.	Cost over/under run (M)	Categorization of Metrics,Results after measurement,Comparision of the results in accordance to planning.
		MAA 2.1.v.	Project Priority (M)	Categorization of Metrics,Results after measurement,Comparision of the results in accordance to planning.
		MAA 2.1.vi.	Net Returns from the Project(O)	Categorization of Metrics,Results after measurement,Comparision of the results in accordance to planning.
		MAA 2.2.	Process Metrics(M)	Categorization of Metrics,Results after measurement,Comparision of

				the results in accordance to planning.
		MAA 2.2.i.	Defect Removal Efficiency (M)	Categorization of Metrics,Results after measurement,Comparision of the results in accordance to planning.
		MAA 2.2.ii.	Review Effectiveness (O)	Categorization of Metrics,Results after measurement,Comparision of the results in accordance to planning.
		MAA 2.2.iii.	Schedule Variation Baseline(M)	Categorization of Metrics,Results after measurement,Comparision of the results in accordance to planning.
		MAA 2.2.iv.	Cost Variation Baseline (M)	Categorization of Metrics,Results after measurement,Comparision of the results in accordance to planning.
		MAA 2.2.v.	Requirement Specificity (M)	Categorization of Metrics,Results after measurement,Comparision of the results in accordance to planning.
		MAA 2.3.	Product Metrics (M)	Categorization of Metrics,Results after measurement,Comparision of the results in accordance to planning.
		MAA 2.3.i.	Mean time between failure(MTBF) (M)	Categorization of Metrics,Results after measurement,Comparision of the results in accordance to planning.
		MAA 2.3.ii.	Minor Defects Left (O)	Categorization of Metrics,Results after measurement,Comparision of the results in accordance to planning.
		MAA 2.3.iii.	Defects reported after release (M)	Categorization of Metrics,Results after measurement,Comparision of the results in accordance to

				planning.
		MAA 2.3.iv.	Availability (M)	Categorization of Metrics,Results after measurement,Comparision of the results in accordance to planning.
		MAA 2.3.v.	Maturity (M)	Categorization of Metrics,Results after measurement,Comparision of the results in accordance to planning.
		MAA 2.4.	People Metrics (M)	Categorization of Metrics,Results after measurement,Comparision of the results in accordance to planning.
		MAA 2.4.i.	Productivity (M)	Categorization of Metrics,Results after measurement,Comparision of the results in accordance to planning.
		MAA 2.4.ii.	Adaptability (M)	Categorization of Metrics,Results after measurement,Comparision of the results in accordance to planning.
		MAA 2.4.iii.	Accuracy (O)	Categorization of Metrics,Results after measurement,Comparision of the results in accordance to planning.
		MAA 2.4.iv.	Innovation (O)	Categorization of Metrics,Results after measurement,Comparision of the results in accordance to planning.
		MAA 2.5.	Defect Density (M)	Categorization of Metrics,Results after measurement,Comparision of the results in accordance to planning.
		MAA 2.6.	Defect Detection Efficiency (M)	Categorization of Metrics,Results after measurement,Comparision of the results in accordance to planning.

		MAA 2.7.	Requirement Stability Index (O)	Categorization of Metrics,Results after measurement,Comparision of the results in accordance to planning.
		MAA 2.8.	System Spoilage (O)	Categorization of Metrics,Results after measurement,Comparision of the results in accordance to planning.
		MAA 2.9.	Maintainability (M)	Categorization of Metrics,Results after measurement,Comparision of the results in accordance to planning.
		MAA 2.9.i.	Effort distribution over activities(analysis design ,implementation, testing) (M)	Categorization of Metrics,Results after measurement,Comparision of the results in accordance to planning.
		MAA 2.9.ii.	Cost of the Project(Analysis, Design, Implementation, testing, etc) (M)	Categorization of Metrics,Results after measurement,Comparision of the results in accordance to planning.
		MAA 2.9.iii.	Effort Distribution over Change Types(Adaptation, error correction, enhancement, others) (M)	Categorization of Metrics,Results after measurement,Comparision of the results in accordance to planning.
		MAA 2.9.iv.	Effort Variation over Change Types (Adaptation, error correction, enhancement, others) (O)	Categorization of Metrics,Results after measurement,Comparision of the results in accordance to planning.
		MAA 2.9.v	Effort Variation over Change Types (Adaptation, error correction, enhancement, others)(O)	Categorization of Metrics,Results after measurement,Comparision of the results in accordance to planning.
		MAA 2.10.	Business Process Metrics (O)	Categorization of Metrics,Results after

				measurement,Comparision of the results in accordance to planning.
		MAA 2.11.	Learning and Growth Metrics (O)	Categorization of Metrics, Results after measurement, Comparision of the results in accordance to planning.

Table 6: Measurement and Analysis Process Areas(MAA)

vii) Reuse Process Areas(REU)

REU 1	Basic Level	REU 2.	Elaborated Level	Documentation
REU 1.1.	Identify Reusable Components	REU 2.1.	Data representation of Components(O)	No of reusable components, characteristic of the reusable components, techniques used in reuse,scope of reuse.
REU 1.2.	Use Stored Components	REU 2.2.	Data Transfer of Components(O)	No of reusable components, characteristic of the reusable components, techniques used in reuse,scope of reuse.
		REU 2.3.	State Persistence of Components(O)	No of reusable components, characteristic of the reusable components, techniques used in reuse,scope of reuse
		REU 2.4.	State Scope of the Component(O)	No of reusable components, characteristic of the reusable components, techniques used in reuse,scope of reuse.
		REU 2.5.	Techniques of failure Component(O)	No of reusable components, characteristic of the reusable components, techniques used in reuse,scope of reuse
		REU	Establish a Reuse	No of reusable

		2.6.	Library(O)	components, characteristic of the reusable components, techniques used in reuse,scope of reuse
		REU 2.6.i.	Add Components(O)	No of reusable components, characteristic of the reusable components, techniques used in reuse,scope of reuse
		REU 2.6.ii.	Delete Components(O)	No of reusable components, characteristic of the reusable components, techniques used in reuse,scope of reuse
		REU 2.6.iii.	Modify Components(O)	No of reusable components, characteristic of the reusable components, techniques used in reuse, scope of reuse
		REU 2.6.iv.	Relate Components(O)	No of reusable components, characteristic of the reusable components, techniques used in reuse, scope of reuse
		REU 2.7	Component Library Search Optimization(O)	No of reusable components, characteristic of the reusable components, techniques used in reuse, scope of reuse

Table 7: Reuse Process Areas(REU)

vii) Quality Control Process Areas(QCO)

QCO 1.	**Basic Level**	**QCO 2.**	**Elaborated Level**	**Documentation**
QCO 1.1.	Management Commitment	QCO 2.1.	Education and Training (M)	Quality Control Plan, Cost and Effort, Tools and Techniques ,Resources,

				Employees,Output Report.
QCO 1.2.	Quality Control	QCO 2.2.	Organization(M)	Quality Control Plan, Cost and Effort ,Tools and Techniques ,Resources, Employees,Output Report.
		QCO 2.2.i.	Quality Improvement teams(QIT)(O)	Quality Control Plan, Cost and Effort, Tools and Techniques ,Resources, Employees,Output Report.
		QCO 2.2.ii.	Investigation and Corrective Action(O)	Quality Control Plan, Cost and Effort ,Tools and Techniques ,Resources, Employees,Output Report.
		QCO 2.2.iii.	Quality Audit(M)	Quality Control Plan, Cost and Effort ,Tools and Techniques ,Resources, Employees,Output Report.
		QCO 2.2.iv.	Awareness(M)	Quality Control Plan, Cost and Effort ,Tools and Techniques ,Resources, Employees,Output Report.
		QCO 2.2.v.	Recognition(M)	Quality Control Plan, Cost and Effort ,Tools and Techniques ,Resources, Employees,Output Report.
QCO 1.2.i.	Employee Involvement	QCO 2.3.	Quality Cost Measurement(M)	Quality Control Plan, Cost and Effort ,Tools and Techniques ,Resources, Employees,Output Report.
QCO 1.2.ii.	Employee Empowerment	QCO 2.4.	Service Culture (M)	Quality Control Plan, Cost and Effort ,Tools and Techniques ,Resources, Employees,Output Report.
QCO 1.3.	Target Identification	QCO 2.5.	Defect Identification(M)	Quality Control Plan, Cost and Effort ,Tools and Techniques ,Resources, Employees,Output Report.
QCO 1.4.	Customer Focus	QCO 2.6.	Defect Prevention(O)	Quality Control Plan, Cost and Effort ,Tools and Techniques ,Resources, Employees,Output Report.
		QCO 2.7.	Defect Control(O)	Quality Control Plan, Cost and Effort ,Tools and Techniques ,Resources, Employees,Output Report.

Table 8: Quality Control Process Areas(QCO)

After the development cycle phases are complete, Quality Control is done. And the software is consulted with the customer to own their views about the software. If the customer is satisfied the expert will proceed forward to check whether any process areas need to be added. If any process area needs to be added, they will add it. But if the customer is not satisfied, the experts take their opinion, and again held a Process Kick off Meeting and thus the development cycle phases are done with.

If any process area needs to be added, they are added in the Database Repository, and the expert had a meeting about the process area under which support set it is needed to be added, and according to expert's decision, the process area is added to one of the support set.

After that it is checked in Product Backlog whether there remains any requirement set, if the requirement set is present, again the experts will held Process Kick Off Expert Meeting, and the same procedure follows.

REQM: Requirement Management

Requirement Management defines a common understanding between the customer and the software project of the customer's requirements that will be addressed by the software project. The Requirement Management process areas are divided into two levels. They are Basic Level, and Elaborated Level. The Basic Level of Requirement Management is defined as REQM 1and the Elaborated Level of Requirement Management is defined as REQM 2.

REQM 1.1: Requirement Gathering

Requirement Gathering is defined as a collection of a set of statements which defines what needs to be provided by a solution to meet the expectations of the sponsor, stakeholders and users. Requirement gathering is very important for producing business requirements and functional requirements. If the requirements are incorrectly, inaccurately or incompletely specified there is little chance of the solution what will be needed.

REQM 1.2: Requirement Analysis

Requirement Analysis defines the process of determining user expectations or customer expectations for a new or modified product. The requirements must be quantifiable, relevant and detailed. It is a very important expect of project management.

REQM 1.3: Identification of Customer Needs

To gain in this competitive edge, it is very important to identify the customer needs. For identification of customer needs, the developer needs to do mainly three things. They are: Communicate with your customers and identify how their needs can be satisfied in a better way, Customers feedback should be recorded and interpreted properly and use the information that is collected from the customer to make important decisions about marketing, buying, merchandising and selling.

If identification of customer needs is done properly it results in attracting more customers compare to other competitors, increases the amount the customers spend on money and persuade your customers to shop with you more regularly.

REQM 2.1: Understanding between Customer and Developer

Understanding between Customer and Developer is very important. The software developers should uncover, understand and specify the user requirements. Misunderstandings of user requirements between software developers and users will cause problems in terms of satisfying user needs, defects, cost and schedule during the software development process. If misunderstanding is created then it should be resolved as soon as possible because the sooner misunderstandings are resolved, better is the ability of developers to build a product that meets user requirements and less is the expense associated with corrections later in the life cycle.

REQM 2.2: Identify Requirement Change

Identifying the need for a requirement change is one of the most challenging aspects of project development, and can significantly impact the project.

REQM 2.3: Bi-directional traceability of requirements

Bi-directional traceability of requirements defines proper direction of the evolving product and indicates the completeness of the subsequent implementation in forward direction and backward traceability helps ensure that the evolving product remains on the correct track with regards to the original and/or evolving requirements (that we are building the product right).

REQM 2.4: Documentation of requirements and change requirements

Requirements are usually written as a means for communication between the different stakeholders. This means that the requirements should be easy to understand both for normal users and for developers. One common way to document a requirement is stating what the system must do. Documentation of the changes in the requirements should also be maintained.

REQM 2.5: Requirements Validation

Requirement Validation is very critical to successful system product development and implementation. Validating requirements means ensuring that the set of requirements is correct, complete, and consistent, a model that satisfies the requirements can be created, and a real-world solution can be built and tested to prove that it satisfies the requirements.

REQM 2.6: Requirements Verification

Requirements Verification defines that each requirement has been satisfied by the software or by the product. Requirement Verification can be done by logical argument, inspection, modelling, simulation, analysis, expert review, test or demonstration.

REQM 3.1: Safety Requirements

Safety Requirements are very necessary to ensure that the system carries out the functions needed to make it acceptably safe. An adequate risk assessment should be carried out to achieve the safety requirements.

Safety Requirements are gathered from previous failure modes of components of a system, any of the other possible layers of protection intended to prevent a failure from turning into an accident, or process involved in the development of a system.

REQM 3.2: Requirement Development

Requirement Development gathers the need of customer and translates into a specification of what the system must do. Requirement development consists mainly of three activities. They are gathering candidate requirements, specifying requirements which is done by committing the gathered requirements to tangible media and analyzing requirements, which is done by looking for commonalties and differences among requirements and breaking them down into their essential characteristics.

REQM 3.3: Requirement Reuse

Software reuse is aimed to significantly increment the quality and productivity of software artifacts. Requirements reuse is an approach to systematically use existent requirements documents to reduce the general effort inside the software life cycle. Requirement reuse improves the requirement engineering by recording the adopted suppositions, made decisions and adopted alternatives for future reference.

REQM 3.4: Identify and track requirement attributes

Requirement attributes are useful pieces of information about individual requirements. Attributes may include dependencies, status, risk level, etc. Each project decides which attributes are beneficial. Requirement attributes allow a team to document, track and use information about requirements more effectively. The mechanism to track attributes varies from project.

REQM 3.5: Trace Requirements

Requirement traceability is the ability to describe and follow the life of a requirement. It traces both in both a forwards and backwards direction (i.e., from its origin, through its development and specification, to its subsequent deployment and use, and through periods of on-going refinement and iteration in any of these phases).Some of requirement traceability techniques are Traceability Matrices/Tables, Unique Identifiers, etc.

REQM 3.6: Requirement Management Tools and Metrics

Requirements metrics, when incorporated in requirements management, assist in analyzing the quality of requirements and identifying the reasons for software reengineering/failure. Requirements metrics define the output measures of the software process. There are several tools and metrics. They are Rational Suite Analyst Studio, RDT 3.0, RTM Workshop 5.0, etc.

PLAN: Planning

Planning establishes reasonable plans for performing the software engineering and for managing the software project. These plans are the necessary foundation for managing the software project. The Planning process areas are divided into two levels. They are Basic Level and Elaborated Level. The Basic Level of Planning is defined as PLAN 1 and the Elaborated Level of Planning is defined as PLAN 2.

PLAN 1.1: Scope of the project

The project scope is the definition of what the project is supposed to accomplish and the budget of both time and money that has been created to achieve these objectives. Best results are achieved when both the definition and the budget are precise and detailed. A good project scope document defines specifically what tasks are to be performed or results delivered. It defines the specific date when these deliverables are due and lists the budget allocated for them.

PLAN 1.2: Define the project life cycle

The Project Life Cycle refers to a logical sequence of activities to accomplish the project's goals or objectives. Regardless of scope or complexity, any project goes through a series of stages during its life.

Project activities must be grouped into phases because by doing so, the project manager and the core team can efficiently plan and organize resources for each activity, and also objectively measure achievement of goals and justify their decisions to move ahead, correct, or terminate.

PLAN 1.3: Project Risks

Project risks include the process of deciding how to approach, plan, and execute risk management activities for a project, determining which risks are likely to affect the project and documenting their characteristics and prioritization of risks for further analysis and detailing their probability of occurrence and impact.

PLAN 1.4: Establish Project Plans

Project plans establishes reasonable plans for performing the software engineering and to form a basis for managing the software project. Project plans typically document project purposes and objectives, broad timeline including project start, finish and major milestones, software budget estimates, perform risk assessment, analysis and mitigation, schedule, resources allocated to the project activities and identify special skills needed to accomplish project task, recognise communication needs and develop an appropriate strategy.

PLAN 1.5: Allocation of roles to project members

The project members should be allocated the different roles that is needed to run successfully a project.

PLAN 2.1: Estimation of effort and cost

Its purpose is estimate the project effort and cost for the work products and tasks based on estimation rationale.

PLAN 2.2: Data Management

Data Management manages the data of a project. Data Management comprises all the disciplines related to managing data as a valuable resource.

PLAN 2.3: Organization Standards and Procedures

Identify the Organization Standards and Procedures which will guide the project's software activities, consistent with the needs of the project.

PLAN 2.4: Project Resources

The purpose of this process area is to plan for the necessary resources to perform the project.

PLAN 2.5: Review Project Plans

Its purpose is to review all plans that affect the project to understand project commitments.

PLAN 2.6: Plan Commitments

The purpose of this process area is to have commitment from relevant stakeholders responsible for performing and supporting plan execution.

PLAN 2.7: Documentation and Record Management

Documentation consists of information or data that can be structured or unstructured and accessed by people in an organization and record management provides evidence of the activities of a given organization’s functioning and policies. Records often have strict compliance requirements regarding their retention, access and destruction, and generally have to be kept unchanged.

PLAN 2.8: Software Architecture Design and Review Process

Transform the software requirements into a software architecture that describes the top-level structure and identifies its major components. Architecture Reviews are an effective way of ensuring design quality and addressing architectural concerns. The principle objectives of a software architecture review are to assess architecture’s ability to deliver a system capable of fulfilling the quality requirements and to identify potential risks.

PLAN 2.9: Progress Review

The main objective is to periodically reviewing the project's progress, performance, and issues.

PLAN 3.1: Stakeholder Involvement

The purpose of this process area is to help satisfy stakeholder concerns and promote transparency is to involve project-affected stakeholders in monitoring the implementation of mitigation measures or other environmental and social programs.

PLAN 3.2: Reconcile work and resource levels

The purpose of this process area is to reconcile the project plan to reflect available and estimated resources.

PLAN 3.3: Joint Technical reviews

The joint technical reviews are planned by the developer and the developer proposes the locations and dates and is approved by the acquirer. These reviews shall be attended by persons with technical knowledge of the software products to be reviewed. The reviews shall focus on in-process and final software products, rather than materials generated especially for the review.

PLAN 3.4: Establishing Software Development Environment and Infrastructure

A suitable supporting Software Development Environment and Infrastructure is required to develop the product because it helps for future growth, use of technologies, tools, hardware and processes.

PLAN 3.5: Training of Developers and Users

Training of developers should be done so that they are able to do their work with latest methods, tools and techniques. Training of users should be done so that they are able to do work with newly available software's.

PLAN 3.6: Technology Management

The purpose of technology management allows organizations to manage their technological fundamentals to create competitive advantage. The role of the technology management function in an organization is to understand the value of certain technology for the organization. Continuous development of technology is valuable as long as there is a value for the customer and therefore the technology management function in an organization should be able to argue when to invest on technology development and when to withdraw.

TEST: Testing

Software testing is any activity aimed at evaluating an attribute or capability of a program or system and determining that it meets its required results. Software testing is done so that the software program/application/product meets the requirements that guided its design and development, works as expected and can be implemented with the same characteristics.

The Testing process areas are divided into two levels. They are Basic Level, and Elaborated Level. The Basic Level of Testing is defined as PLAN 1, and the Elaborated Level of Testing is defined as PLAN 2.

TEST 1.1: Test Cases

A test case in software engineering is a set of conditions or variables under which a tester will determine whether an application or software system is working correctly or not.

TEST 1.2: Software Coding and Unit Testing

The goal of the coding is to implement the design in the best possible manner. The coding activity affects testing. During coding, it should be kept in mind that the programs should not be constructed so that they are easy to write, but in a manner that they are easy to read and understand. Unit testing is a software development process in which the smallest testable parts of an application, called units, are individually and independently scrutinized for proper operation. Unit testing is often automated but it can also be done manually

TEST 1.3: Integration Testing

Integration testing is a logical extension of unit testing. Integration testing is a software development process which program units are combined and tested as groups in multiple ways. Integration testing can expose problems with the interfaces among program components before trouble occurs in real-world program execution.

TEST 1.4: System Testing

System Testing defines the testing of the behaviour of the whole software/system as defined in software requirements specification (SRS). System testing is done after integration testing is complete. System Testing validates an application accuracy and completeness.

TEST 1.5: System Integration Testing

System integration testing verifies that a system is integrated to any external or third-party systems defined in the system requirements.

TEST 2.1: Requirement Testing

Requirement testing is testing the requirements whether they are feasible or not. Because a project depends on a number of factors like time, resources, budget etc. Before start working on a project it's important to test the requirement.

TEST 2.2: Performance Testing

Performance testing is the best way to give your product a competitive edge. Performance testing can verify that a system meets the specifications claimed by its manufacturer or vendor. The process can compare two or more devices or programs in terms of parameters such as speed, data transfer rate, bandwidth, throughput, efficiency or reliability.

TEST 2.3: Regression Testing

Regression testing is the process of testing changes to computer programs to make sure that the older programming still works with the new changes. Regression testing is a normal part of the program development process.

TEST 2.4: Manual Support Testing

Manual Support Testing involves all the functions performed by people in preparing data for and using data from, automated applications. The manual support testing verifies that the manual support procedures are documented and complete, determine that manual support responsibility has been assigned, determine that the manual support people are adequately trained and determine that the manual support and the automated segment are properly interfaced.

TEST 2.5: Acceptance Testing

Acceptance is the final stage of testing that is performed on a system prior to the system being delivered to a live environment.

TEST 2.6: Build Test Teams

A test team need to be built so that the testing processes are done in an effective way and the members to the team should be selected based on experienced in testing field.

TEST 2.7: Test Estimation and Planning

The purpose of test estimation and planning is to estimate the cost and effort required during testing and according to that the plan for testing is decided.

TEST 2.8: Test Results Reporting

The purpose of Test Results Reporting is to report about the results of all testing's that was carried out during the development of the software product.

TEST 2.9: Alpha test and Beta test (α test and β test)

Alpha testing is simulated or actual operational testing by potential users/customers or an independent test team at the developers' site. Alpha testing is often employed for off-the-shelf software as a form of internal acceptance testing, before the software goes to beta testing.

Beta testing comes after alpha testing and can be considered a form of external user acceptance testing. Versions of the software, known as beta versions, are released to a limited audience outside of the programming team. The software is released to groups of people so that further testing can ensure the product has few faults or bugs.

TEST 2.10: Test Reports and Acceptance

The Software Acceptance Test Report is primarily addressed to the software developers and summarizes the tests performed and their results.

TEST 3.1: Test Metrics

Test Metrics are a standard of measurement. Test Metrics provide the efficiency and effectiveness of several software development activities. They are gathered and interpreted through test effort. It provides an objective measurement of the success of a software project. The Test Metrics are

TEST 3.1.i.Number of test cases planned

The purpose of number of test cases planned defines the total number of test cases planned for a particular project.

TEST 3.1.ii.Number of test cases failed

The purpose of number of test cases failed defines the total number of test cases failed out of the total number of test cases planned for a particular project.

TEST 3.1.iii Number of test cases passed

The purpose of number of test cases passed defines the total number of test cases passed out of the total number of test cases planned for a particular project.

TEST 3.1.iv.Number of bugs opened

The purpose of number of bugs opened defines the total number of bugs found in a particular unit time.

TEST 3.1.v.Number of bugs closed per unit of time

The purpose of number of bugs closed per unit of time defines the total number of bugs solved per unit of time out of the total number of bugs found.

TEST 3.1.vii.Test Effectiveness ratio

Test Effectiveness Ratio is defined as the number of statements executed by the test data to the total number of executable statements.

TEST 3.2: Test Environments

A testing environment is a setup of software and hardware on which the testing team is going to perform the testing of the newly built software product. The testing setup need to be built on both the sides i.e. client and server.

TEST 3.3: Test Process Improvement

Test process improvement (TPI) provides with a framework to streamline and improve your testing process. In testing, the challenge is to find the fastest and most cost-effective way to get the best testing results. TPI works by finding strong and weak points in the testing cycle via gap analysis.

TEST 3.3.i. Analytical risk-based testing strategies

Analytic approach is used to identify risks and test them.

TEST 3.3.ii.Realistic objectives for testing

The purpose of realistic objectives for testing defines that the testing and test cases should be done and generated based on realistic objective of the project.

TEST 3.3.iii.Continuous Test Process Improvement

The purpose of continuous test process improvement is that the testing processes should be continuously updated with the complexity of the project and for fulfilling the market demand.

TEST 3.3.iv.Trained and Certified Test Teams

The teams which are responsible for testing should undergo training in order to update with new technologies and tools of testing and also certified and experienced people in testing should be in the test team.

TEST 3.3.v. Distribute Testing work intelligently

The purpose of distributing testing work intelligently means that the different testing works of the project should be distributed properly among the test teams according to experience and knowledge of testing.

TEST 3.4.Test System Design and Implementation

The purpose of test system design and implementation is to test the system design of the software project and then test whether the design will be implemented in real world or not.

TEST 3.5.Test Automation

The purpose of test automation is to automate the different types of testing which is done manually.

QAS: Quality Assurance

Quality Assurance is a process centered approach to ensuring that a company or organization is proving the best possible products or services. The purpose of Quality Assurance is to provide management with appropriate visibility into the process being used by the software project and of the products being built. Quality Assurance is an integral part of most software engineering and management processes.

QAS 1.1.Measuring Quality

Quality of software is measured based on different circumstances. There are different techniques to measure the quality of software. It may be based on completeness of the features of the software, usability of the users in the software, use of metrics, processes, and bug detection.

QAS 1.2.Reducing Project & Product Risk

Project Risk which affects project resources and schedule. The Project Manager (PM) should keep a watchful eye on such risk. PM regularly judges the progress, forecasts schedule risk, and initiates proper actions to overcome this risk. Product Risk affects quality or performance of the software being developed.

QAS 2.1.Standards Procedures, Techniques and Tools

Every software development company must establish standards and procedures to aid in the development of software, because these standards are the framework for which software evolves. Techniques are used to perform product evaluation and process monitoring and tools are developed for problem, change, and configuration management along with many different testing tools.

QAS 2.1.i.Process Quality Assurance (Measurement and Feedback)

The purpose of Process Quality Assurance is measured by the features of the software, metrics, bugs, etc and this is checked through feedback.

QAS 2.1.ii.Quality Assurance Team

The Quality Assurance Team (QAT) implements a consistent and repeatable approach for quality assurance review of technology projects. The quality assurance team uses a popular tool commonly called PDCA (Plan, Do, Check, Act) to evaluate and determine excellence and provide improvement.

QAS 2.1.iii.Continually improve Processes

The processes need to be continually improved because the processes reflects the quality of a product and as the market demand changes frequently ,the processes need to be improved according to market demand so that the product is accepted in the market.

QAS 2.2.Software Configuration Management and Change Control

Software Configuration Management a discipline for managing and controlling the evolution of a system during development, in production, during maintenance & use and through retirement.

QAS 2.2.i.Configuration Identification

The purpose of this process area is to identify and uniquely name the components. Give a version number to each of the configuration item and then establish ‘configuration baselines’ by identifying the structure of how configuration items interconnected to form the system. Revision is done to configuration items and the change is traced.

QAS 2.2.ii.Configuration Control

The purpose of Configuration Control is to manage the changes to an established system baseline. It assigns change roles and responsibilities, process change requests, specify required change activities, track change history and approvals and establish new system baselines.

QAS 2.2.iii.Configuration Status Accounting

The purpose of Configuration Status Accounting is to documentation in a paper trail or electronic trail showing the history of configuration items and changes to them. It includes dates, component identifications, versions, baseline descriptions, change details and information about the people involved. It also reports about the report queries if an automated configuration management system is used.

QAS 2.2.iv.Configuration Auditing

Configuration Auditing is also known as Configuration Evaluation. The purpose of Configuration Auditing is to evaluate the effectiveness of the configuration management system. It reviews the trail generated by Status Accounting and verifies that the system's current configuration matches what the configuration management system claims.

QAS 2.3.Change Control

Change control is the process used to request, review, plan, approve, and implement changes to a system.

QAS 2.4.Formal Technical Reviews

Formal technical review are software quality assurance activity performed by software engineers to uncover errors in function, logic or implementation of the software, verify that the software meets its requirements, ensure that the software has been developed according to standards, achieve uniform software and make projects manageable.

QAS 2.5.Statistical Quality Assurance

Statistical Quality Assurance reflects the growing trend throughout industry to become more quantitative about quality. In order calculate the statistical quality assurance, data's are collected and categorized. An attempt is made to trace each defect to its underlying cause. Once the causes are identified, the defects are corrected.

QAS 2.6.Process Management

The Process Management provide leadership to software dependent organizations through proven, process-focused methods for improving product costs, schedule and quality.

QAS 2.6.i.Clarify current knowledge of the process

If the team members of a particular project don't have proper knowledge of the processes that the project is undergoing, it needs to clarify the current knowledge of the process so that they can properly work on the projects, which results in a quality product.

QAS 2.6.ii.Identify the process needed to be improved

The processes that reflect the quality of a product and also reflect with market demand need to identify and also check whether they need to be improved.

RIMA: Risk Management

Risk Management is concerned with all activities that are performed to reduce the uncertainties associated with certain tasks, or events. The Risk Management process areas are divided into two levels. They are Basic Level, and Elaborated Level. The Basic Level of Risk Management is defined as RIMA 1, and the Elaborated Level of Measurement and Analysis is defined as RIMA 2.

RIMA 1.1: Identification of Risk

In project management, risk identification begins at the earliest stages of a project and continues throughout the project life cycle. Project risks can include unknown issues and variability in cost, effort, timing, and benefits in relation to a specific project. Project risks can be either internal risks or external risks. Internal risks usually arise from a failure of the project organization or resources to achieve their expected performance. External risks usually arise from market activity, fiscal policies affecting currency, inflation and taxation environmental factors or social impact.

RIMA 1.2: Risk Analysis

Risk analysis helps in identifying, assessing and reducing risks to an acceptable level. It defines and controls threat and vulnerabilities and implements risk reduction measures.

RIMA 2.1: Planning of risk

Risk Planning helps to identify risks and implement a plan to reduce them. Using risk plan, the risks can be monitor and control effectively, increasing the chances of achieving success. The risk plan categorizes and prioritizes each risk, determine the likelihood of the risks occurring and identify the impact on the project if risk does occur.

RIMA 2.2: Tracking of Risk

During the life of the project, risks and associated actions need to be monitored. The Risk Tracking Database will be the principal repository of risk escalation history.

RIMA 2.2.i: Controlling of Risks

Track the effects of the risks identified and manage them to a successful conclusion. During risk tracking and control the project leaders must ensure that risks stay visible to the project and business management teams.

RIMA 2.2.ii: Risk Communication

After identifying that the project involves risk, communicating those risks is a special skill. People often share information about project risks ineffectually or not at all. As a result, their projects suffer unnecessary problems and setbacks that proper communication may have avoided. Communicate about project risks early and often for successful output of a project.

MAA: Measurement and Analysis

The Measurement and Analysis provides a solid infrastructure for implementing a measurement program. Measurement and analysis involves gathering quantitative data about products, processes and projects and analyzing that data to influence actions and plans.

Measurement and analysis activities characterizes or gain understanding of processes, products, resources and environments, evaluate to determine the status with respect to plans, predict by understanding relationships among processes and products so the values observed for some attributes can be used to predict others and improve by identifying roadblocks, root causes, inefficiencies, and other opportunities for improvement.

The Measurement and Analysis process areas are divided into two levels. They are Basic Level, and Elaborated Level. The Basic Level of Measurement and Analysis is defined as MAA 1, and the Elaborated Level of Measurement and Analysis is defined as MAA 2.

MAA 1.1: Target Response Time

Target Response Time is defined as the time between receipt of the call and the time that a Micro Support Team member begins working on the problem.

MAA 1.2: Length of Source Code

Length of source code is a software metric used to measure the size of a software program by counting the number of lines in the text of program’s source code.

MAA 1.3: Duration of Testing Process

The time taken by a tester to test from the start of a project to the end of a project and it is measured by elapsed time in hours.

MAA 1.4: Time a Programmer spends on a project

The time spent by a programmer to build the project and it is measured by the months worked

MAA 1.5: Usability

Usability is the ease of use and learnability of a human-made object.

MAA 2.1: Project Metrics

The project metrics are used by project manager to adapt project workflow and technical activities. The project metrics guide adjustments to work schedule to avoid delays and assess product quality on an ongoing basis. Some of the project metrics are

MAA 2.1.i. Size of the Project

Size of the project is calculated in terms of LOC, Function Point, etc.

MAA 2.1.ii Cost of the Project

Cost of the project defines the total cost required to complete the project.

MAA 2.1.iii.Time over/under run

Time over/under run defines whether the status of the project is completed within time or whether it is taking more time to complete the project in accordance with the planning.

MAA 2.1.iv.Cost over/under run

Cost over/under run defines whether the project will be able to complete the project within budget or whether the project will not be able to complete under planned budget.

MAA 2.1.v. Project Priority

The projects are given priority based on the importance and urgency of the customers.

MAA 2.1.vi.Net Returns from the Project

The total calculation of the project after it is delivered to the customer.

MAA 2.2: Process Metrics

Process metrics are collected across all projects and over long periods of time. They are used for making strategic decisions. The intent is to provide a set of process indicators that lead to long term process improvement. The only way to know how/where to improve any process is to measure specific attributes of the process, develop a set of meaningful metrics based on this attributes and used the metrics to provide indicators that will lead to a strategy for improvement.

MAA 2.2.i.Defect Removal Efficiency

DRE is a measure to detect defects before delivery. It is calculated as a percentage of the defects identified and corrected internally with respect to the total defects in the complete project life cycle. Thus, DRE is the percentage of bugs eliminated by reviews, inspections, tests etc.

MAA 2.2.ii.Review Effectiveness

Review Effectiveness is a metrics that evaluates the quality / effectiveness of reviews. Review Effectiveness is equal to ratio of Defects found in Reviews by Total Defects found in project.

MAA 2.2.iii.Schedule Variation Baseline

Any deviation from the baseline plan of a project, measured by comparing budgeted cost of work scheduled with budgeted cost of work performed.

MAA 2.2.iv.Cost Variation Baseline

The deviation from actual baseline of the cost as initially planned by the developers and stakeholders.

MAA 2.2.v.Requirement Specificity

Requirement Specificity is measured by the ratio of number of requirements for which reviewers had identical interpretation to total number of functional requirements plus total number of non-functional requirements.

MAA 2.3: Product Metrics

Product metrics assesses the state of the project, track potential risks, uncover problem areas, adjust workflow or tasks and evaluate team's ability to control quality.

MAA 2.3.i.Mean time between failure (MTBF)

MTBF is an expression of reliability that attempts to predict the life expectancy of a device in hours.

MAA 2.3.ii.Minor Defects Left

Minor Defects Left defines that a defect is not removed at a given time and it can be removed later on at approximately same cost or penalty.

MAA 2.3.iii.Defects reported after release

Defects reported after release defines the number of defects found in an software after it is being outsourced to the customers.

MAA 2.3.iv.Availability

Availability defines that the data can be easily accessed anytime.

MAA 2.3.v.Maturity

Maturity defines the state of art of fully developed.

MAA 2.4.People Metrics

People Metrics defines the contribution of human resource towards productivity of software product, adaptability to new tools, techniques and methods, accuracy in doing a particular work and their contribution towards new ideas.

MAA 2.5: Defect Density

Defect density is a measure of the total known defects divided by the size of the software entity being measured.

$$\frac{\text{Number of Known Defects}}{\text{Size}}$$

The Number of Known Defects is the count of total defects identified against a particular software entity, during a particular time period.

MAA 2.6: Defect Detection Efficiency (DDE)

Defect Detection Efficiency (DDE) is the number of defects detected to the total number of defects.

MAA 2.7: Requirement Stability Index

A requirement stability index (RSI) is a metric used to organize, control, and track changes to the originally specified requirements for a new system project or product.

MAA 2.8: System Spoilage

System spoilage is defined as the effort spent in fixing faults divided by the total project effort.

MAA 2.9: Maintainability

Maintainability is the degree to which a program is open to change i.e. mean time to change, change request to new version (analysis, design, etc) and cost to correct.

MAA 2.9.i. Effort distribution over activities (analysis design, implementation, testing)

Effort distribution over activities defines distribution of effort across maintenance activities.

MAA 2.9.ii.Cost of the Project (Analysis, Design, Implementation, testing, etc)

Cost of the project defines distribution of cost across maintenance activities.

MAA 2.9.iii.Effort Distribution over Change Types (Adaptation, error correction, enhancement, others)

Effort Distribution over Change Types defines distribution of effort over types of maintenance.

MAA 2.9.iv.Effort Variation over Change Types (Adaptation, error correction, enhancement, others)

Effort Variation over Change Types defines similarity/difference in effort distribution in different types of maintenance.

MAA 2.10.Business Process Metrics

Business Process Metrics are for internal business processes of the organization which are carried out to meet end-to-end customer needs. The processes may be strategic or support ones. Metrics for support processes can be best defined by people handling the work processes, and may be generic across business set-ups, whereas strategic process metrics may be uniquely identified as by nature a strategy will have a factor of differentiation over the earlier ways of doing work.

MAA 2.11.Learning and Growth Metrics

Continuous learning and adaptation is a major ingredient to growth. Software Managers need to gather and analyze metrics that guide them on the strengths and weaknesses of their people and skills. Intranet and knowledge portals established in organizations are a move towards facilitating this aspect of the organization. Some of learning and growth metrics are the average number of knowledge claims contributed per person, patents and copyrights registered by the organization in a year, hits per knowledge claim and effectiveness of the knowledge transfer program.

REU: Reuse

Reuse is the recovery of materials and products for the same or a similar end use. The reuse process areas are divided into two levels. They are Basic Level and Elaborated Level. The Basic Level of Reuse is defined as REU 1 and the Elaborated Level of Reuse is defined as REU 2.

REU 1.1.Identify Reusable Components

Identifying and extracting the reusable components from already developed and existing software systems or legacy systems can save the cost of developing the software.

REU 1.2.Use Stored Components

Components those are extracted for reuse are used to match the functionality with the requirements of the developing software, and if it is matched the components are reused.

REU 2.1 Data representation of Components

In order to exchange data between two components, it is necessary that the two components agree on the same data type.

REU 2.2.Data Transfer of Components

Components encounter mismatches in the direction of data flow as well as the sequence of data transfer. Components may either push data or may pull data. A component that is developed for the push mode will not work in the pull mode.

REU 2.3.State Persistence of Components

Some components are stateless whereas other components retain their state. State persistence may be an assumption made by an client (calling) component about the server(supplier) component. This assumption produces unexpected results if the server turns out to be stateless.

REU 2.4.State Scope of the Component

State scope defines the extent of the exposure of the component’s internal state.

REU 2.5.Techniques of failure Component

The technique to report component failures may differ from one component to another. Gluing together components with different failure mechanisms causes mismatches.

REU 2.6.Establish a Reuse Library

Reusable components are stored together in a library.

REU 2.6.i.Add Components

The purpose of Add Components refers to adding new components to the set of selected components.

REU 2.6.ii.Delete Components

The purpose of Delete Components refers to deleting components from the set of selected components.

REU 2.6.iii.Modify Components

The purpose of Modify Components refers to modifying components from the set of selected components in case of functionality and structure.

REU 2.6.iv.Relate Components

The purpose of relate components refers to relating different components in functionality and structure from the set of components.

REU 2.7.Component Library Search Optimization

The purpose of Component Library Search Optimization refers to searching of components of different functionality and structure in the library of components.

QCO 1.1.Commitment Management

Commitment Management is a radical new approach to requirement definition, negotiation and contract management. It is a discipline that accelerates high quality, low cost business transactions and relationships through applying zero-defect quality principles that enhance competitiveness and profitability.

QCO 1.2.Quality Control

The purpose of Quality Control to build a quality council so that the employees can involve in quality for the product and also innovate new ideas.

QCO 1.2.i.Employee Involvement

Employee involvement is a process for empowering employees to participate in managerial decision-making and improvement activities appropriate to their levels in the organization.

QCO 1.2.ii.Employee Empowerment

Employee Empowerment means that all employees feel that they have the responsibility and authority to participate in decision making and problem solving in their appropriate operating levels.

QCO 1.3.Target Identification

The act of determining the nature of a target, including the goals to achieve. Based on it the areas are prioritized.

QCO 1.4.Customer Focus

Customer Focusing is a technique for identifying the customers, products/services, quality characteristics, and performance measures which are most important to the business unit. The technique can be used successfully to identify key factors to include in a Quality Policy Statement. A Customer Focusing session is designed for staff working together in a particular business or business unit.

QCO 2.1.Education and Training

Education and training are a vital part for quality control. Training is one of the key elements of total quality in which many people are involved, so that the success of the implementation depends directly on how well it is done. Local trainers need to have their skills developed in order to customize and deliver materials relevant to their audience. The trainer's job is to make the material come alive for those being trained by relating it directly to experience and needs. The materials itself falls into two distinct parts, education in the principles of total quality-prevention, understanding, customer requirements, internal customers, teamwork, etc and training in tools and techniques for problem solving.

QCO 2.2.Organization

Using the company's structure, an organization for quality should be developed based on objectives, delegation, accountability, achievement measurement and efficient communication.

QCO 2.2.i.Quality Improvement teams (QIT)

Quality Improvement teams (QIT) solve problems identified by goal settings and address local issues. The characteristics of QIT are intradepartmental, membership is open based on knowledge of the operation, the leader should know the business and membership is finite.

QCO 2.2.ii.Investigation and Corrective Action

To reach the targets, the QIT should set a system of procedures, forms and documentation.

QCO 2.2.iii.Quality Audit

The Quality Audit is conducted by the QIT or by the senior members. The goals of quality audit are verify reported quality improvements, verify the achievement of goals and confirm that improvements match targets.

QCO 2.2.iv.Awareness

Awareness is done about continuous communication about quality status and advances and employs communication tools that are compatible with the company's culture.

QCO 2.2.v.Recognition

Recognition of achievement of individuals and groups are made public .Financial reward is also given to individuals and groups.

QCO 2.3.Quality Cost Measurement

Quality cost measurement should be done in the following ways. They are review the literature on quality costs or consult others in similar industries that are using the same tool, select one organizational unit of the company to serve as a pilot site, discuss the objective of the study with the key people in the organization, collect whatever cost data are conveniently available from the accounting system and present the cost results to management along with the results of a demonstration quality improvement project.

QCO 2.4.Service Culture

Service culture drives a service mindset in product and service focused organisations. A Service Culture indicates that, for everyone in the provider organization, Customer satisfaction is the top priority and that service provider activities demonstrably contribute to the business objectives of the Customer.

QCO 2.5.Defect Identification

Defect Identification is the maintenance task during which defects are identified in the deployed application and its documentation.

QCO 2.6.Defect Prevention

The purpose of Defect Prevention is to identify the causes of defects and analyze the defects that were encountered in the past and take specific actions to prevent the occurrence of those types of defects in the future.

QCO 2.7.Defect Control

The purpose of Defect Control is identify the defects and get it corrected before it leads more defects and failure and also to prevent defects from which conditions the defects had arised from previous conditions.

Chapter 6

Future Scope and Conclusion

The survey carried out in this thesis made many things crystal clear that small and medium scale organizations are using either agile methodologies or any variant of agile i.e combination of RUP and Agile processes, combination of Waterfall and Agile. The SPI Expert Model in SMEs integrated processes those are recognized as important and efficient to the development of software with agile methodologies. The model is capable of delivering a number of customized process models according to the resources, need and choice of the software organization. As the software professionals can select processes they want to use from Basic and Elaborated level in given process support set, makes SPI Expert model more customized .The addition of all basic set processes to the chosen processes Elaborated set of processes make the model more complete. This model can be made more beneficial by adding dependencies among different processes.

References

[1].A. Classe, TickIT update. *Software Quality Management*, no.19, 10–11, 1993.

[2].A. Cockburn, *Agile Software Development*, Boston: Addison Wesley, 2002a

[3].A. Cockburn, *Surviving Object Oriented Projects: A Manager's Guide*, Addison Wesley Longman, 1998.

[4].A. Dorling, "SPICE: Software process improvement and capability determination," *Software Quality Journal*, vol. 2, no. 4, pp. 209–224, Dec 1993.

[5]A. Finkelstein, J. Kramer, B. Nuseibeh, *Software Process Modelling and Technology*,Research Studies, 1994.

[6].Advanced Development Methods, *Controlled chaos: living on the edge*, 1996.http://www.controlchaos.com/old-site/ap.htm

[7].A. Cockburn | Crystal Clear.
http://alistair.cockburn.us/crystal/books/ alistairsbooks.html)

[8].B. Curtis, M. Kellner, J. Over, "Process modeling". *Communications of the ACM* ,vol. 35,no. 9 pp. 75-90,September,1992.

[9].B. Curtis, W. Hefley and S. Miller, *People Capability Maturity Model*. Addison-Wesley: Boston, MA, USA, 2001.

[10].B. Franken, *How to Explain Six Sigma by Using the Profit Triangle*. Retrieved May, 2007 from www.iSixSigma.com/library/content/.

[11].B. McFeeley,*IDEAL SM: A User's Guide for Software Process Improvement*, Software Engineering Institute, Carnegie Mellon University, February, 1996.

[12].C. Larman,*Agile and Iterative Development:A Manager's Guide.*Addison-Wesley,2004.

[13].C. Larman and V. Basili,"A History of Iterative and Incremental Development,"IEEE Computer, vol.36, no.6, pp.47-56, 2003

[14].Commander Airforce Materiel Command, "Acquisition Software Development Capability Evaluation," *AFMC PAMPHLET*, vol. 1,15th June, 1994.

[15].C. Bao-Ling ,”System Security Engineering Capability Maturity Model”, *Journal of the Harbin Institute of Technology* , vol. 35,no. 3,pp.293-297,2002.

[16].F. Coallier, R. McKenzie, J.F. Wilson and J. Hatz, *The Trillium Model*, Version 3, Bell Canada, December, 1994.

[17].F. Niessink and H.V. Vliet, ”Towards Mature IT Services,” *Software Process Improvement and Practice*, vol.4, no.2,pp-55-71,1998.

[18].G. Eckes, *The Six Sigma Revolution, How General Electric and Others Turned Process into Profits*. USA: John Wiley & Sons, Inc., 2001.

[19].G. Brue and R.G. Launsby , *Design for Six Sigma*, McGraw-Hill Professional,2003.

[20].I. Jacobson,*Object Oriented Software Engineering* .New York:Addison-Wesley,1994.

[21].INCOSE Capability Assessment Working Group, *Systems Engineering Capability Assessment Model*, Version 1.50, June 1996.

[22].ISO *Quality Systems: Model for Quality Assurance in Design/ development, Production, Installation and Servicing*. ISO-9001, International Standards Organization, 1987.

[23].J.A. Highsmith,*Adaptive Software Development :A Collaborative Approach to Managing Complex Systems*.New York,NY,Dorset House Publishing,2000.

[24].J. C. Derniame, B. A. Kaba, D. Wastell, *Software Process: Principles, Methodology and Technology*. Lecture Notes in Computer Science 1500,Springer, 1999.

[25].J. Haungs , ”Pair Programming on the C3 project, ”*Computer*, vol. 34,no. 2,pp.118-119,2001.

[26].J. Stapleton,*Dynamic Systems Development Method-The method in practice*.Addison-Wesley,1997.

[27].J. Gremba and C. Myers, *The IDEAL(SM) Model: A Practical Guide for Improvement* ,US:SEI, Carnegie Mellon University,1997.

[28].K. Beck, “Embracing Change with Extreme Programming,” *IEEE Computer*, vol. 32,no. 10,pp. 70-77,October,1999.

[29].K. Beck, *Extreme programming explained: Embrace change*, Reading Mass, Addison-Wesley, 1999.

[30].K. Benali, J. C. Derniame, “Software processes modeling: what, who, and when”. *Proceedings of the Second European Workshop on Software Process Technology* ,September, 1992.

[31].K. Schwaber and M. Beedle, *Agile Software Development With Scrum*, Upper Saddle River, NJ: Prentice Hall, 2002.

[32].K. Schwaber,*Agile Project Management with Scrum*,Microsoft Press,Redmond,2004.

[33].L.G. Jones, *Software Process Improvement and Product Line Practice: CMMI and the Framework for Software Product Line Practice*, US: SEI, Carnegie Mellon University, 2002.

[34].M. Pomeroy-Huff ,Robert Cannon ,Timothy A. Chick ,Julia Mullaney and William Nichols, *The Personal Software Process SM (PSP SM) Body of Knowledge*, Version 2.0, , Software Engineering Institute, Carnegie Mellon University, August 2009.

[35].Martin and N. James, *“Overview of the EIA 632 Standard :Process for Engineering a System,” Eighth Annual International Symposium of INCOSE*, Vancouver, British Columbia, July 26-30, 1998.

[36].M.C. Paulk, B. Curtis, M.B. Chrissis, and C.V. Weber, “Capability Maturity Model, Version 1.1,” *IEEE Software*, vol. 10, no. 4, pp. 18-27, July 1993.

[37].M.C. Paulk,” How ISO-9001 compares with the CMM,” *IEEE Software*, pp.74-83.1995.

[38].M.Konrad,M.B. Chrissis,J. Ferguson, S. Garcia, B.Hefley, D.Kitson and M. Paulk,”Capability Maturity Modelling SM® at the SEI”,*Software Process Improvement and Practice*,vol. 2,no. 1,pp.21-34,1996.

[39].M. Poppendieck and T. Poppendieck,*Lean Software Development: An Agile Toolkit*,Addison-Wesley.2003.

[40].N.Davis and J. McHale,”Relating the Team Software ProcessSM(TSP SM) to the Capability Maturity Model ® for Software(SW-CMM®) With Strategy and Overview by Watts S. Humphrey ”,Carnegie Mellon Software Engineering Institute, Technical Report CMU/SEI-2002-TR-008 ESC-TR-2002-008,March,2003.

[41].P. Kruchten,” Agility with the RUP”, *Cutter IT Journal*,vol.14 ,no.12, pp.27-33,2001.

[42].P.Kruchten ,*The Rational Unified Process :an Introduction.*Addison-Wesley,2000.

[43].P. Kuvaja,Software Process Assessment and Improvement:the BOOTSTRAP Approach,UK:Blackwell Publishers,Oxford,1994.

[44].P. Vakaslahti," Process Improvement Frameworks- a Small Case Study with People Capability Maturity Model", *Software Process Improvement and Practice,* vol. 3,no.4 ,pp.225-234,December 1997.

[45]. P.Allen, M. Ramachandran and H. Abushama," PRISMS: an Approach to Software Process Improvement for Small to Medium Enterprises", in *Proc. of the Third International Conference On Quality Software (QSIC'03)*,pp. 1-4,March 2003.

[46].Pino, J. Francisco, García, Felix; and M. Piattini, "Software process improvement in small and medium software enterprises: a systematic review", *Software Quality Control*,vol.16,no.2, pp.237-261, September 1985.

[47].R. Singh,"International Standard ISO/IEC 12207 Software Life Cycle *Processes",Software Process Improvement and Practice*,vol. 2,no. 1,pp.35-50,1996.

[48].R. R. Wright, "Process Standards and Capability Models for Engineering Software Intensive Systems," *CROSSTALK :The Journal of Defense Software Engineering*,pp.7-11,October 1998. cross5talk2.squarespace.com/.../**1998**/**1998**10/**1998**10-**Wright**.pdf.

[49].R. Bate et. al., *A Systems Engineering Capability Maturity Model*, Version 1.1, Software Engineering Institute, Carnegie Mellon University, November 1995.

[50].Rout and P. Terence," SPICE: A Framework for Software Process Assessment," *Software Process Improvement and Practice*, vol. 1, no. 1, pp. 57-66, 1995.

[51].S.R. Palmer and J.M. Felsing, *A Practical Guide to Feature-Driven Development* .Upper Saddle River,NJ: Prentice Hall,2002.

[52].S. Zahrin, *Software Process Improvement: Practical Guidelines for Business Success*. New Jersey: Addison Wesley, 1998.

[53].S. Arnold and H.W. Lawson,"Viewing Systems from a Business Management Perspective:The ISO/IEC 15288 Standard ,"*System Engineering*,vol.7,no.3,pp.229-242,2004.

[54].S.M. Hwang ,"Quality Metrics for Software Process Certification based on K-Model"*in Proc. IEEE 24th International Conference on Advanced Information Networking and Applications Workshops*,pp. 827-830,2010.

[55].*System Security Engineering Capability Maturity Model® SSE-CMM®,Model Description Document*,Version 3.0,15th June ,2003.

[56]T. Doran,"IEEE 1220: For Practical Systems Engineering",*Computer*,pp. 92- 94,May,2006. **ieee**xplore.**ieee**.org/iel5/2/34216/01631953.pdf

[57].TickIT, *TickIT Guide to Software Quality Management System Construction and Certification using ISO 9001:1994.* Issue 1.0, DISC TickIT Office, November, 1995.

[58].*Trillium:Model for Telecom Product Development & Support Capability*, Release 3.0, Bell Canada, December 1994.

[59].V.M. Hasse,"BOOTSTRAP: Fine Tuning Process Assessment," IEEE Software, vol.11, no.4, pp.25-37, 1994.

[60].Watts S. Humphrey, *Introduction to the Team Software Process.* Reading, MA: Addison-Wesley Publishing Company, Inc., 2000.

[61].Watts S.Humphrey, *Managing the Software Process*. Addison-Wesley,1989.

[62].Watts S. Humphrey, "The Personal Software Process: Status and Trends", *IEEE Software*, vol.17, no.6, pp.71-75, Dec 2005.

[63].Werth, Laurie Honour**, "**Introduction to Software Process Improvement", Carnegie Mellon University, CMU/SEI-93-EM-8.

[64].http://asq.org/learn-about-quality/malcolm-baldrige-award/overview/overview.html

[65].http://en.wikipedia.org/wiki/Capability_Maturity_Model_Integration

[66].http://en.wikipedia.org/wiki/Malcolm_Baldrige_National_Quality_Award

[67].http://www.ambysoft.com/essays/spiTips.html

[68].www.abelia.com/**498pdf**/**498-STD**.**PDF**

[69].www.jfpconsulting.co.uk/**Standards**.htm

APPENDIX A: QUESTIONNAIRES

APPENDIX A.1: Questionnaires for Senior Management

APPENDIX A.2: Questionnaires for Project Manager

APPENDIX A.3: Questionnaires for Team Leader

APPENDIX A.4: Questionnaires for Software Engineer (Team Member)

APPENDIX A.5: Questionnaires for the need of a model

APPENDIX A.1: Questionnaires for Senior Management

Name of the Company:____

Address of the Company____

1.1. Enterprise ownership:

(a) Foreign investment (b) Local

(c)State owned company

Year of foundation..

1.2. Number of employees in software activity: …………………………………………………

1.3. Which following quality management systems/ principles are applying (or in process of development) in your company?

(a) None (b)TQM (Total quality management)

(c) ISO 9000 (d) CMM (Capacity maturity model)

(e)CMMI (Capability Maturity Model Integration) (f)SPICE

(g)BOOTSTRAP (h)K-Model

(i)Six Sigma (j)TickIt

(k)Trillium (k) Others: ....

If ans is (d),then do you use variant of CMM,if yes please write the name?.....................................

1.4. What are the software project’s category your company undertakes?

(a)Commercial Software (b)Outsourcing

(c)Build-to-order software (d)Others(please specify.....................)

1.5.Number of people involved in project team(Approximately)

(a)In the whole project life cycle: ………………

(b) In the planning stage: ………………………..

(c)In the analysis and design stage: …………….

(d) In the coding stage: ………………………….

(e) In the testing stage: ………………………….

(f)In the deploying stage: ………………………

(g)In the maintenance stage: ……………………

2.1How is the organization best described?

(a) Commercial off the self (e.g. shrink-wrap or custom installation of enterprise solutions such as SAP or ORACLE)

(b)Contracted new development (e.g. for use in particular product lines or other novel point solutions)

(c)In house or proprietary development

(d)Defense Contractor

(e)Other Government Contractor

(f)Other Government Agency

(g)Others

2.2.What is the primary focus of the organization work?

(a)Product or System Development

(b)Maintenance or sustaintment

(c)Acquisition

(d)Service Provision

(e) Others

2.3.In what **sectors/fields** has your company developed software products?

(a)Agriculture/ Forestry and Sea–food	(b) Government administration
(c)Commercial and Service	(d) Manufacturing
(e)Construction	(f) Medicine
(g)Education and Training	(h) Telecommunication
(i)Entertainment	(j)Tourism
(k)Finance & Accounting	(l)Transportation

(m)Others (please specify) ... ...

2.4. Who are your company's major clients?

(a)Foreign organizations	(b)Local organizations
(c)America (North)	(d) Government organizations
(e)America (South)	(f)Private organizations
(g)Asia	(h)Individuals
(i)Europe	(j)India

(k)Others (please specify) ... ...

2.5. Did the organization adopt any of the following system development methods in the projects?

(a)Structured approach (SDLC) (b) Object – oriented

(c)Prototyping, interactive (d) Rational Unified Process (RUP)

(e)Rapid application development (RAD)

(f) Agile software development processes

(g)Others (please specify) ……………………………………

2.6.Most of the projects in your company completes:

(a)Ahead of schedule (b) Over 30% behind schedule

(b)On time (d) Over 50% behind schedule

2.7. The real cost spent for most of the projects in your company

(a)Ahead of budget (b)Over 30% cost overrun

(c)Within planned budget (d) Over 50% cost overrun

3. What kind of engineering is major part's of organization's work?

(a)Software Engineering

(b)Hardware Engineering

(c)Systems Engineering

(d)Design Engineering

(e)Test Engineering

(f)Others

4. Which statistical methods are used in the organization's process performance modelling?

(a) Regression analysis predicting continuous outcomes (e.g., bivariate or multivariate linear regression or non-linear regression)

(b) Regression analysis predicting categorical outcomes (e.g., logistic regression or loglinear models)

(c) Analysis of variance (e.g., ANOVA, ANCOVA or MANOVA)

(d) Attribute SPC charts (e.g., c, u, p, or np)

(e) Individual point SPC charts (e.g., ImR or XmR)

(f) Continuous SPC charts (e.g., XbarR or XbarS)

(g) Design of experiments

(h)Others (Please specify...)

5. Which of the following visual display techniques are used to communicate the results of the organization's analyses of process performance baselines?

(a) Box Plots

(b)Histograms

(c)Scatter Plots or multivariate charting

(d)Pareto charts,pie charts or bar charts

(e)Mosaic Charts for categorial data

(f)Others(Please specify..)

(g)None of the above

(h) Don't know

6.Which of the technical challenges the projects sometimes face?

(a) Initial project requirements are not well defined

(b) Requirements change significantly throughout the life of the projects

(c) There is little or no precedent for the kind of work we are doing

(d) Significant constraints are placed on product quality attributes (e.g. reliability, scalability, security, supportability, etc.)

(e) The size of the development effort is large

(f) The technology needed for the projects is not mature

(g) There are extensive needs for interoperability with other systems

(h) Insufficient resources (e.g. people, funding) are available to support the projects

(i) Insufficient skills and subject matter expertise are available to support the projects

(j) Other (Please specify..)

(h) None of the above

7. Which best describe the organization's documentation of its process performance and quality measurement results

(a) In the organization's PIIDs,

(b) Measurement repository

(c) Process asset library

8.Which requirement management tools are used in your organization?

(a)Rational Suite Analyst Studio(Use Case Modeling)
(b)RDT 3.0
(c)RTM Workshop 5.0
(d)Telelogic DOORS
(e) Omni Vista OnYourMark Pro
(f)Starbase Caliber RM
(g)Others (Please specify..)

9.Which process modelling tools are used in your organization?
(a)BizAgi Process Modeler
(b)Questetra BPM Suite
(c)Tibco Business Studio
(d)Process Maker
(e)Open ModelSphere
(f)Others(Please Specify...)

10.Which software configuration management(SCM) tools are used in your organization?
(a)GNU Arch
(b)ClearCase
(c)Concurrent Version System(CVS)
(d)Perforce
(e)Serena ChangeMan DS
(f)SubVersion
(g)Others(Please Specify..)

11. Which software development tools are used in your organization?
(a)QuickLicense 6
(b)Click Install 3.0
(c)AppProtect 2.0

(d)WinA&D 7
(e)DocProtect 2.0
(f)OfficeProtect 1.0
(g)WebActivation 1.1
(h)AirLicense 1.0
(i)Others(Please specify..)

12.Which cost models are used in your organization?
(a) Advanced Missions Cost Model
(b) COCOMO Software Cost Model
(c)Cost Estimating Guidelines
(d)Cost Spreading Calculator
(e)Learning Curve Calculator
(f)DSN Cost Estimating Cost Model
(g)Others(Please specify)

13. Do your Organization follows any agile methodologies ?
(a)Yes (b)No

13.1.Which methodologies are being followed in your organization?

(a)Extreme Programming
(b)Scrum
(c)Crystal Methodologies
(d)Adaptive Software Development
(e)Rational Unified Process
(f)Evolutionary Project Management EVO
(g)Feature Driven Development
(h)Dynamic System Development Method
(i)Lean Development

If ans is (c),please specify which methodology...

13.2.Which Extreme Programming tools are used in your organization?
(a)Ant
(b)JUnit
(c)Cactus
(d)Others(Please specify...)

13.3. Which Scrum tools are used in your organization?
(a)Kunagi
(b)ScrumDo
(c)Express
(d)Scrumy
(e)Pango Scrum
(f)Scrumpy
(g)Agilito
(h)Banana Scrum
(i)Sprintometer
(j)Others(please specify...)

13.4. Which RUP tools are used in your organization?
(a)Rational Rose
(b)Others(Please Specify..)

13.5. Which Lean Development tools are used in your organization?
(a)Quality Function Deployment
(b)Design Structure Matrix
(c)Visual Control Board
(d)Others(Please Specify..)

13.6. Which Feature-Driven Development tools are used in your organization?
(a)CASE Spec
(b)TechExcel DevSuite
(c)FDD Tools
(d)FDD Viewer

14. Which project planning tools are used in your organization?
(a)Brainstorming
(b)Fishbone/Ishikawa Diagrams
(c)Critical Path Analysis Flow Diagram
(d)Gantt Charts
(e)Microsoft Projects
(f)Others (Please Specify..)

15.What are the software features used to support project management?

1.	Work breakdown structure	Never			Always	Not Available	
		1	2	3	4	5	0

2.	Gannt Chart	Never			Always	Not Available	
		1	2	3	4	5	0

3.	CPM Chart(Network)	Never			Always	Not Available	
		1	2	3	4	5	0

4.	PERT Chart	Never			Always	Not Available	
		1	2	3	4	5	0

5.	Budget Management	Never			Always	Not Available	
		1	2	3	4	5	0

6.	Handling Conflicts	Never			Always	Not Available	
		1	2	3	4	5	0

7.	Progress Tracking	Never			Always	Not Available	
		1	2	3	4	5	0

8.	Weekend Modifier	Never			Always	Not Available	
		1	2	3	4	5	0

9.	Overbooking of resources	Never			Always	Not Available	
		1	2	3	4	5	0

10.	Workforce Assignment	Never			Always	Not Available	
		1	2	3	4	5	0

11.	Others	Never			Always	Not Available	
		1	2	3	4	5	0

16.To which processes risk analysis is done?

(a)Requirement Management

(b)Software Project Planning

(c)Software Project Tracking and Oversight

(d)Software Configuration Management

(e)Software Quality Assurance

(f)Organization Process Focus

(g)Organization Process Definition

(h)Peer Reviews

(i)Software Subcontract Management

(j)Training Program

(k)Integrated Software Management

(l)Software Product Engineering

(m)Intergroup Coordination

(n)Software Quality Management

(o)Quantitative Process Management

(p)Defect Prevention

(q)Technology Management

(r)Supplier Agreement Management

(s)Resource Management

17. Do the employees are knowledgeable about current processes of the organization?

(a)Yes (b)No

18. If yes, whether they are confidence and knowledgeable enough for a change?

(a)Yes (b)No

19. Do there is a sharing of knowledge and experience among the employees in the organization?

(a)Yes (b)No

20.Do the employees are able to identify strengths and weakness of the current processes?

(a)Yes (b)No

21.Do the organization envisioned how the improved software process might look like?

(a)Yes (b)No

22.Do the envisioned process is actually implemented, accepted and executed by managers and employees?

(a)Yes (b)No

23. Do the envisioned process becomes the current process in the organization?

(a)Yes (b)No

24. Is there shortages of software engineers and management positions?

(a)Yes (b)No

25 Is there lack of commitment and continuity by the leaders responsible for software improvement?

(a)Yes (b)No

26. Is there will and discipline among the employees for a change in the organization?

(a)Yes (b)No

27. Is there a budget for increasing staff skills in software engineering, management and continuous improvement?

(a)Yes (b)No

28.Do the senior management and respected people are involved in process improvement?

(a)Yes (b)No

29.Do the changes in the organization is brought about gradually and systematically?

(a)Yes (b)No

30.Do the employees involved in software process improvement are able to relate to the realities and needs of software development?

(a)Yes (b)No

30.Do the organization focuses on processes ,people and technology simultaneously?

(a)Yes (b)No

31.On what basis the company does the recruitment?...

32. Have the gaps and desired improvements been identified in order of priorities?

(a)Yes (b)No

33. Are clear and specific goals and priorities for SPI established?

(a)Yes (b)No

34. Do people know why the change has been envisioned, and what's their role in the entire Software Process Improvement (SPI) journey?

(a)Yes (b)No

Comment:

34. Is a change agent with organization-wide visibility and credibility across the organizational layers identified?

(a)Yes (b)No

35. Are mechanisms defined to ensure management involvement on an on-going basis?

(a)Yes (b)No

36. Are measures to gauge progress of the SPI defined and made known to all the stakeholders?

(a)Yes (b)No

37. Is a training plan defined for the people to acquire the new skills?

(a)Yes (b)No

38. Are development staff and line managers involved in improving the identified processes and executing the change?

(a)Yes (b)No

39. Are rewards and recognition planned for those contributing to the Software Process Improvement (SPI) journey?

(a)Yes (b)No

40. Are individual objectives formally linked to the SPI objectives?

(a)Yes (b)No

41. Is there a forum for exchange of ideas and suggestions for SPI?

(a)Yes (b)No

42. Have workshops aiming at providing guidance for SPI been planned for all levels for organizations, including the senior management?

(a)Yes (b)No

43. Does a formal project plan exist for SPI with milestones, timelines, and review mechanisms?

(a)Yes (b)No

44. Are review mechanisms defined to assess the progress and impact of change?

(a)Yes (b)No

45. Have all the key risk factors been identified and addressed through mitigation and contingency plans?

(a)Yes (b)No

APPENDIX A.2: Questionnaires for Project Manager

Name of the Company:____
Address of the Company____

1. Which leadership style is followed in your company?

(a) Democratic, all the members participate in making decision process

(b)To consult project team before project manager makes decision

(c)Project manager makes all decision without consulting

(d)People-oriented leadership

(e)Work-oriented leadership

(f) Others…………………………………………………………………………..

(g) Unclear

2. The level of resources available in the planning stage is:

2.1.	Manpower	Very Low			Very High	No Opinion	
		1	2	3	4	5	0

2.2.	Time	Very Low			Very High	No Opinion	
		1	2	3	4	5	0

2.3.	Budget	Very Low			Very High		No Opinion
		1	2	3	4	5	0

2.4.	Infrastructure	Very Low		Very High		No Opinion	
		1	2	3	4	5	0

2.5.	Others	Very Low		Very High		No Opinion	
		1	2	3	4	5	0

3. Identifying project infrastructure

3.1. How did the project select the team members?

....

....

....

4. Did the personnel of sale and marketing department participate in project team? How are their roles in planning?

....

....

....

5. Are changes to the software commitments made with the involvement and agreement of the affected group?

(a) Yes (b) No

6. Are software project commitments and changes to commitments made to individuals and groups external to the organisation, reviewed with senior management according to a documented procedure?

(a) Yes (b) No

7. The software process database of the organization is used for software planning and estimating.

(a) Yes (b) No

8. Appropriate software engineering methods and tools are established and integrated into the software process of each project.

(a) Yes (b) No

9. Do you have a pay-for-performance policy that rewards good performing employees?

(a) Yes (b) No

10. Do you conduct surveys to compare how your compensation policies fair against your competitors'?

(a) Yes (b) No

11. Do your organization go for reuse?

(a) Yes (b) No

12. Do your organization contains project repository?

(a) Yes (b) No

13. Whether the employees are given training on how to reuse repository material?

(a) Yes (b) No

14. Do your organization communicate laws and regulations governing the work environment to the workforce?

(a) Yes (b) No

15. Adequate desktop space for the tools that are used in performing tasks and the work products that are being produced and manipulated.

(a) Yes (b) No

16. The work environment is periodically analyzed to identify changes or resources that could improve work performance.

(a) Yes (b) No

17. Whether critical dependencies between engineering groups are identified, negotiated, and tracked according to a documented procedure?

(a) Yes (b) No

18. Does the management plan reflects the organization goals and structure?

(a) Yes (b) No

19. Are there documents containing the information involved in the quantitative management?

(a) Yes (b) No

20. Is there a quality group that checks if plans and procedures of the process management are fulfilled?

(a) Yes (b) No

21. Is there a team that is trained to analyze the process quantitatively?

(a) Yes (b) No

22. Are there any documents that register the changes?

(a) Yes (b) No

23.In average, how many people involved in a project in your company?....................................

24. Give your opinion of the following statements

24.1.	Quality of software product is mainly depending on programmer's ability rather than have or not any quality management system.	Strongly Agree		Neutral		Strongly Disagree
		5	4	3	2	1

24.2.	Usability (is measure of re-usability of code)	Very Important		Average		Very Important
		5	4	3	2	1

APPENDIX 1.3: Questionnaires for Team Leader

Name of the Company:____
Address of the Company____

1. In managing the project team,how much time Project Leaders in Software Companies spend on following works?

1.1.	Team Management (staffing;team structure;team communication;team building;conflict resolution;performance appraisal and training)	Very Much				Very Little
		5	4	3	2	1

1.2.	Communication management (Speed up the relationship between project team and stakeholders)	Very Much				Very Little
		5	4	3	2	1

1.3.	Risk management (Identify and avoid or mitigate the project risks throughout the life cycle of a project)	Very Much				Very Little
		5	4	3	2	1

1.4.	Quality management (Implementing the activities, techniques and strategy to ensure that all project	Very Much				Very Little
		5	4	3	2	1

	activities and work products comply with all relevant standards, procedures and requirements)					

1.5.	Time management (Applying the processes and techniques of time management to ensure the timely completion of the project)	Very Much			Very Little	
		5	4	3	2	1

1.6.	Cost management (Estimating budget; managing and controlling cost in order to ensure that the project is completed within the approved budget)	Very Much			Very Little	
		5	4	3	2	1

2. Your project estimates time and cost based on:

(a) Experiences ☐ (b) Specific software ☐

3. Please rank the following project evaluation criteria in order of importance in your project.

(a) Completion on planned time

(b) Completion within project

(c) Completion at desired level of quality

(d) Customer satisfaction

(e) Capability improvement for company

(f) Others

4. The project management methodology/tools

4.1. Did your project team apply any project management methodology in this project?

(a) Yes (b) No
(Name of the method..)

4.2. This methodology has been formalized by:

(a)A local company (b)A foreign company

(c)Your company (in-house) (d) Others(.......................................)

4.3. Did your project team use any software to manage this project?

(a) Yes (b) No

(Name of the Software...)

4.4. This software has been developed by:

(a)A local company (b) A foreign company

(c)Your company (in-house) (d) Others(..)

5.

5.1.	In the planning stage, delegates of the company functional departments (sales, marketing, networking, database, etc.) participated actively as project members.	Strongly Disagree			Strongly Agree		No Opinion
		1	2	3	4	5	0

5.2.	In the planning stage, delegates of the client's functional departments participated actively as project members.	Strongly Disagree				Strongly Agree	No Opinion
		1	2	3	4	5	0

5.3.	In the planning stage there was no conflicting objectives between the project team and the customer.	Strongly Disagree				Strongly Agree	No Opinion
		1	2	3	4	5	0

6. Please, tick out the methods that are followed in your company

6.1. The common methods used to communicate with client in this project are:

(a)Informal Meeting

(b)Formal Meeting

(c)Email

(d)Telephone

(e)Others

6.2. The common methods used to communicate within the project team in this project are:

(a)Informal Meeting

(b)Formal Meeting

(c)Email

(d)Telephone

(e)Others

7. How the performances of the team members are rated in the project?

....

....

....

8. Is the project manager kept informed of the software project's status and issues?

(a) Yes (b) No

9. The involvement of client/ end-user in the planning stage

9.1.	Level of involvement of customer/ user in planning stage	Very Low			Very High	No Opinion	
		1	2	3	4	5	0

9.2. If this involvement is "low", pls. explain the reasons

....

....

10. Is the documented software development plan used and maintained as the basis for tracking the software project?

(a) Yes (b) No

11. What corrective actions are taken when targets of the software plan are not achieved?

(a) Yes (b) No

12. Is the size of the software work products (or size of the changes to the software work products) tracked, and corrective actions taken, as necessary?

(a) Yes (b) No

13. Whether a documented plan is used to communicate intergroup commitments and to coordinate and track the work performed?

(a) Yes (b) No

14. Is there any plan to prevent defects?

(a) Yes (b) No

15. Is defect documentation collected?

(a) Yes (b) No

16. Is a plan used for the Process Control Management?

(a) Yes (b) No

17.

17.1.	Do you establish and document criteria for evaluating potential suppliers?	Always	Usually	Sometimes	Rarely if ever	Never	Do not Know	Not Apply

17.2.	Does the organization has written policy or guideline for planning the supplier agreement management process?	Always	Usually	Sometimes	Rarely if ever	Never	Do not Know	Not Apply

18. Are customer requirements grouped/coordinated into a set of requirements?

(a) Yes (b) No

19. Is the responsibility for requirements development split between the acquirer and developer, and who is responsible for which levels?

(a) Yes (b) No

20. Is the project's software development plan revised according to a documented procedure?

(a) Yes (b) No

21. Is the software development plan reviewed at each revision?

(a) Yes (b) No

22. Is the software development plan managed and controlled?

(a) Yes (b) No

23. Are formal reviews to address the accomplishments and results of software projects conducted at selected project milestones according to a documented procedure?

(a) Yes (b) No

24.What aspects of the activities are reviewed?

(a) Approved plan (Baseline).

(b) Risk management.

(c) Quality assurance (QA).

(d) Change management process.

(e) Configuration management.

(f) Subcontract management.

(g) Others, please specify...

25.How often are these reviews conducted?

(a) reviews once a week.

(b)reviews once a fortnight.

(c) reviews once a month.

(d) reviews on ad hoc basis.

(e) reviews at other intervals.(plz specify..)

(f) reviews more frequently.

26. Is there any plan informing of the day-by-day activities and a detailed explanation for each one?

(a) Yes (b) No

27. Are there policies that follow and analyze the process management?

(a) Yes (b) No

28. Do you evaluate field employees on a regular basis?

(a) Yes (b) No

29.	Does the project follows a written management policy to manage software requirements	Yes	No	Doesn't Apply	Don't Know

30.	Are requirements and their assignations managed and documented through the whole life cycle of the project?	Yes	No	Doesn't Apply	Don't Know

APPENDIX A.4: Questionnaires for Software Engineers
Name of the Company:____
Address of the Company____

1.1 Does the software engineering group conduct periodic internal reviews to track technical progress against the software development plan?

(a) Yes (b) No

1.2. Does the software engineering group conduct periodic internal reviews to track plans, against the software development plan?

(a) Yes (b) No

1.3. Does the software engineering group conduct periodic internal reviews to track performance against the software development plan?

(a) Yes (b) No

1.4 Does the software engineering group conduct periodic internal reviews to track technical issues against the software development plan?

(a) Yes (b) No

1.5. Who conducts these reviews?

(a)Software Engineer(Team Members)

(b)Team Leader

(c)Project Manager

(d)Software Metric Group

(e)Software Process Improvement Team

(f)Others(Please Specify..)

2. What type of measurements are made and used to determine the status of the software tracking and oversight activities?(in terms of activities)

(a) Actual effort and remaining effort (in hours)

(b) Number of tasks completed and not completed.

(c) Milestones completed and not completed.

(d) Actual task start and finish dates.

(e) Earned value analysis (% complete).

(f) Pert Analysis

(g) Others, please specify...

3. What type of measurements are made and used to determine the status of the software tracking and oversight activities?(in terms of resource information)

(a) Actual effort and remaining hours.

(b) Actual costs.

(c) Critical computer resources.

(d) Others, please specify.

4. What type of measurements are made and used to determine the status of the software tracking and oversight activities?(in terms of performance and quality information)

(a) Current mean time between failures (MTBF).

(b) Lines of code generated.

(c) Number of reworks and adherence to quality criteria.

(d) Number of changes put through change control.

(e) Actual size of work products.

(f) Number of component parts completed.

(g) Test results.

(h) Number of issues raised/discussed.

(i) Number of action items.

(j) Others, please specify...

5. Does your project's software processes are developed by tailoring the standard software process of the organization and whether this is performed according to a documented procedure?

(a) Yes (b) No

6. The software development plans of the projects, which describe the tailoring of the standard software process, are developed and revised according to a documented procedure.

(a) Yes (b) No

7.Each software project is managed in accordance with the project's defined software process.

(a) Yes (b) No

8.1. The software is designed according to the project's software process.

(a) Yes (b) No

8.2. The software is developed according to the project's software process.

(a) Yes (b) No

8.3. The software is documented according to the project's software process.

(a) Yes (b) No

8.4. The software is verified according to the project's software process.

(a) Yes (b) No

9.1. The developed software code is maintained, according to the a defined process.

(a) Yes (b) No

9.2. The developed software code is documented according to the a defined process.

(a) Yes (b) No

9.3. The developed software code is verified according to the a defined process.

(a) Yes (b) No

10. It is verified that the software design is implemented correctly and complete in the software code.

(a) Yes (b) No

11. Whether the software engineering group and the other engineering groups participate with the customer and end users, as appropriate, to establish the system requirements?

(a) Yes (b) No

12. Are changes in software made to improve the processes of the whole organization?

(a) Yes (b) No

14. Are technological changes planned?

(a) Yes (b) No

15. Are technological changes related to organization strategies?

(a) Yes (b) No

16. Is verification performed early and continuously throughout the project life-cycle?

(a) Yes (b) No

17. Are validation processes used to ensure that received products/services fulfil their intended use?

(a) Yes (b) No

18. Has the project defined the degree to which validation is required, both early in the project definition and when products/systems are received?

(a) Yes (b) No

APPENDIX A.5: Questionnaires for Need of Software Process Model (Senior Management)

Name of the Company:____
Address of the Company____

1. Whether your organization is able to produce software products within time?

(a) Yes (b) No

2.Do your organization follow any specific software process model?

(a) Yes (b) No

If yes,please specify the name of the process model____________________________

3. Did the software process model used in the organization helped the project teams brings clarity to the process?

(a) Yes (b) No

4. The detailed software process model has brought clarity in roles and responsibilities.

(a) Yes (b) No

5. The software process model has helped or will help in improving communication between the team members.

(a) Yes (b) No

6. Agreed software process model have indirectly helped or will help to improve the project KPIs.

(a) Yes (b) No

7. It is difficult to get agreement on the software process model from all stakeholders.

(a) Yes (b) No

8. The use of software process model reduces some of the project-related issues.

(a) Yes (b) No

List of Publications

Nomi Baruah and Ashima," A Rigorous Study of Software Process Improvement Models in Software Industry", *in Proceedings International Conference on Competitiveness & Innovativeness in Engineering, Management & Information Technology(ICCIEMI-2012)*,29th January,2012.

Nomi Baruah and Ashima,"A Rigorous Study of Software Process Improvement Models in Software Industry", *International Journal of Research in Engineering & Applied Science*, vol.2, no.2, pp. 1144-1153, February,2012.

Nomi Baruah and Ashima, "Contribution of Software Process Improvement Approaches for Small and Medium Scale Enterprises", *International Journal of Computing and Corporate Research*,vol.2,no.2,March,2012.

Nomi Baruah and Ashima," A Survey of the Use of Agile Methodologies in Different Indian Small and Medium Scale Enterprises (SMEs)", *International Journal of Computer Applications*, vol. 48, June 2012. (Status: Communicated)

Nomi Baruah and Ashima,"A Survey of the Use of Project Planning Tools in Different Indian Small and Medium Enterprises (SMEs)", *in Proceedings International Conference on Advanced Computing Technologies*, 8th-9th June, 2012.(Status: Accepted)

Nomi Baruah and Ashima,"A Survey of the Use of Project Planning Tools in Different Indian Small and Medium Enterprises (SMEs)",*International Journal of Computers & Technology*,2012. (Status: Accepted)

Nomi Baruah and Ashima,"SPI Expert in SMEs",*in Proceedings of CSI 6th International Conference on Software Engineering*,2012.(Status: Communicated)